



СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY

Ю. В. Вайнштейн, Т. Г. Пенькова, В. И. Вайнштейн

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

УЧЕБНОЕ ПОСОБИЕ



**ИНСТИТУТ КОСМИЧЕСКИХ
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Министерство науки и высшего образования Российской Федерации
Сибирский федеральный университет

Ю. В. Вайнштейн, Т. Г. Пенькова, В. И. Вайнштейн

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Учебное пособие

Красноярск
СФУ
2019

УДК 510.6(07)
ББК 22.122я73
В145

Рецензенты:

Л. Ф. Ноженкова, доктор технических наук, профессор, зав. отделом прикладной информатики Института вычислительного моделирования СО РАН;

Г. М. Рудакова, кандидат физико-математических наук, профессор кафедры информационно-управляющих систем Сибирского государственного университета науки и технологий имени академика М. Ф. Решетнева

Вайнштейн, Ю. В.

В145

Математическая логика и теория алгоритмов : учеб. пособие / Ю. В. Вайнштейн, Т. Г. Пенькова, В. И. Вайнштейн. – Красноярск : Сиб. федер. ун-т, 2019. – 110 с.

ISBN 978-5-7638-4076-6

Изложены темы, традиционно изучаемые в курсе математической логики и теории алгоритмов: алгебра логики и исчисление высказываний, логика и исчисление предикатов, формальные аксиоматические теории, теория алгоритмов и теория вычислительной сложности.

Предназначено для студентов направления подготовки 09.03.04 «Программная инженерия». Также будет полезно студентам направлений 09.03.02 «Информационные системы и технологии», 27.03.03 «Системный анализ и управление».

Электронный вариант издания см.:
<http://catalog.sfu-kras.ru>

УДК 510.6(07)
ББК 22.122я73

ISBN 978-5-7638-4076-6

© Сибирский федеральный университет, 2019

ОГЛАВЛЕНИЕ

Введение	4
1. Алгебра логики	6
1.1. Введение в алгебру логики	6
1.2. Формулы алгебры логики	21
1.3. Законы алгебры логики	29
1.4. Стандартные формы представления формул алгебры логики	33
1.5. Функционально полные системы элементарных булевых функций	39
Практические задания	47
2. Формальные теории.....	59
2.1. Исчисление высказываний как формальная теория	59
2.2. Исчисление предикатов как формальная теория	65
2.3. Автоматическое доказательство теорем. Принцип резолюций	74
Практические задания	80
3. Теория алгоритмов	88
3.1. Основные понятия теории алгоритмов	88
3.2. Машина Тьюринга	92
3.3. Вычислимые по Тьюрингу функции	96
Практические задания	98
Заключение.....	107
Библиографический список	108

ВВЕДЕНИЕ

Дисциплина «Математическая логика и теория алгоритмов» изучает алгебру логики и исчисление высказываний, логику и исчисление предикатов, формальные аксиоматические теории, а также основные понятия теории алгоритмов и теории вычислительной сложности.

В учебном плане образовательной программы 09.03.04 «Программная инженерия» СФУ дисциплина «Математическая логика и теория алгоритмов» обеспечивает освоение общепрофессиональных компетенций: ОПК-1 «Владение основными концепциями, принципами, теориями и фактами, связанными с информатикой»; ДОПК-1 «Способность использовать основные законы естественно-научных дисциплин в профессиональной деятельности». Таким образом, курс позволяет формировать математическую и информационную культуру студента, обеспечивает приобретение систематизированных знаний, умений и навыков в области математической логики и теории алгоритмов, изучения ее основных методов, механизмов их развития и применения для решения научных и практических задач в области будущей профессиональной деятельности.

В результате изучения дисциплины студент должен знать специальную терминологию, которая является частью языка современной математики, основы логики высказываний, логики предикатов, теории алгоритмов и теории вычислительной сложности алгоритмов, принципы построения формальных теорий, способы применения логических функций. Также студент должен уметь проводить формально-логические построения на основе теории и формул математической логики, строить доказательства теорем в классических теориях исчисления высказываний и предикатов, строить и анализировать алгоритмы для решения прикладных задач. Кроме того, студент должен получить практические навыки использования языка математической логики для представления знаний о предметных областях, построения математических теорий как аксиоматических теорий и интерпретации формул теории и её моделей.

Логика известна человечеству с древнейших времён и представляет собой искусство правильно рассуждать. Термин «логика» (от греч. *logos*) означает *рассуждение, слово, понятие, разум*. Признанным ос-

нователем науки о логике является древнегреческий философ Аристотель, заложивший теоретически основы логики как науки. Основная функция логики состоит в построении последовательности вывода одних утверждений из других, т. е. в выявлении формальных условий правильного мышления. На протяжении истории сфера конкретных интересов логики существенно менялась. Существует логика неформальная, формальная, диалектическая, логика исследования и др.

В учебном пособии представлены разделы, классически отнесенные к формальной, а именно математической логике: алгебра логики, формальные аксиоматические теории и теория алгоритмов. В первой главе рассмотрены формулы алгебры логики, законы алгебры логики, стандартные формы представления формул алгебры логики и функционально полные системы элементарных булевых функций. Во второй главе представлены классические исчисления математической логики: исчисление высказываний и исчисление предикатов, а также метод резолюций для автоматического доказательства теорем. Последняя глава курса посвящена теории алгоритмов. В ней представлена одна из моделей алгоритмов – машина Тьюринга, а также затронуты вопросы оценки вычислительной сложности алгоритмов. В конце каждой главы приведены практические задачи. Также пособие снабжено библиографическим списком.

Настоящее учебное пособие главным образом предназначено для студентов, изучающих математическую логику и теорию алгоритмов, имеющих базовую подготовку по дискретной математике, математическому анализу, информатике и программированию.

В дополнение к материалу, изложенному в первой главе пособия, рекомендуется ознакомиться с основными понятиями, изучить принятую в области алгебры логики терминологию, рассмотреть основные приложения алгебры высказываний к логико-математической практике [2; 4; 8; 12; 15; 19; 21; 22]. Для получения более подробной информации о формальных теориях рекомендуется обратиться к источникам [5; 11; 15; 20; 24]. Подробнее познакомиться с теорией вычислимых функций и конкретными вычислимыми моделями (машина Тьюринга и вычислимые функции) можно в [6; 9; 13; 14; 17; 25; 26]. Для активного самостоятельного решения задач по математической логике и теории алгоритмов рекомендуется использовать задачки [3; 7; 10; 16].

1. АЛГЕБРА ЛОГИКИ

1.1. Введение в алгебру логики

Рассмотрим двухэлементное множество B и двоичные переменные, принимающие значения из B . Его элементы часто обозначают 0 и 1, однако они не являются числами в обычном смысле. Наиболее распространенная интерпретация двоичных переменных: «да» – «нет», «истинно» (И) – «ложно» (Л). Поэтому будем считать, что $B = \{0, 1\}$, рассматривая 0 и 1 как некоторые формальные символы.

Алгебра, образованная двухэлементным множеством $B = \{0, 1\}$ вместе со всеми возможными операциями на нем, называется *алгеброй логики*.

Функцией алгебры логики (логической функцией) от n -переменных называется n -арная операция на множестве $\{0, 1\}$. Логическая функция $f(x_1, x_2, x_3, \dots, x_n)$ – это функция, принимающая значения 0, 1. Множество всех логических функций обозначается P_2 , множество всех логических функций n переменных – $P_2(n)$.

Исходным понятием математической логики является «высказывание». *Высказыванием* называется повествовательное предложение, о котором можно сказать в данный момент, что оно истинно или ложно, но не то и другое одновременно. Логическим значением высказывания являются «истина» или «ложь».

Пример

Повествовательное предложение «3 – это простое число» является истинным высказыванием, а «3,14... – рациональное число» – ложным; «Юго-восточный берег озера Виви является географическим центром России» – истинным, а «Красноярск – столица России» – ложным, «Число 8 делится на 2 и на 4» – истинным, а «Сумма чисел 2 и 3 равна 8» – ложным и т. п.

Такие высказывания называют *простыми*, или *элементарными*. При формальном исследовании сложных текстов понятие «простые высказывания» замещают понятием «*пропозициональные переменные*» (от лат. *propositio* – предложение), которое обозначают прописными буквами латинского алфавита « A », « B », « C » и т. д.

«*Истинность*» или «*ложность*» предложения есть истинностное значение высказывания. Каждому высказыванию сопоставляется

переменная, равная 1, если высказывание истинно, и равная 0, если высказывание ложно.

Пример

1. Если $A :=$ «3 – это простое число», то $A = 1$.
2. Если $B :=$ «Красноярск – столица России», то $B = 0$.
3. Если $C :=$ «Москву основал Юрий Долгорукий», то $C = 1$.
4. Если $D :=$ «В прямоугольном треугольнике квадрат гипотенузы равен сумме квадратов катетов», то $D = 1$.
5. Если $E :=$ « $2 \cdot 2 = 5$ », то $E = 0$.

Следующие утверждения не являются высказываниями.

1. $a + b = 2$.
2. Математика – интересный предмет.

Примечание: символ « $:=$ » означает, что пропозициональной переменной, стоящей слева, необходимо присвоить значение высказывания, стоящего справа от символа.

Высказывания, которые получаются из простых предложений с помощью грамматических связок «не», «и», «или», «если..., то...», «... тогда и только тогда, когда...» и т. п., называют сложными, или составными. Для обозначения грамматических связок вводят символы, которые называют логическими (или пропозициональными) связками.

Обычно рассматривают следующие логические связки:
отрицание (читается «НЕ», обозначается « \neg »),
конъюнкция (читается «И», обозначается « \wedge »),
дизъюнкция (читается «ИЛИ», обозначается « \vee »),
импликация (читается «ЕСЛИ... ТО...», обозначается « \rightarrow »),
эквивалентность (читается «...ТОГДА И ТОЛЬКО ТОГДА, КОГДА...», обозначается « \leftrightarrow »).

Для построения сложных пропозициональных высказываний используют вспомогательные символы «(», «)» – скобки.

Пример

- Если $A_1 :=$ «3 – вещественное число», то $A_1 = 1$.
 Если $A_2 :=$ «3 – целое число», то $A_2 = 1$.
 Если высказывание «3 – вещественное или целое число», то его формула записывается $(A_1 \vee A_2) = 1$.

Правила построения сложных высказываний в виде последовательности пропозициональных переменных, логических связок

и вспомогательных символов определяют возможность формального описания любого текста. При этом высказывания, из которых делают вывод новых высказываний, называют *посылками*, а получаемое высказывание – *заключением*.

Множество пропозициональных переменных $T = \{A, B, C, \dots\}$ с заданными над ними логическими операциями $F = \{ \neg; \wedge; \vee; \rightarrow; \leftrightarrow \}$ формируют *алгебру высказываний*, т. е. $A_v = \langle T; F \rangle$.

Логические операции

Из простых высказываний с помощью операций над высказываниями или логических операций (связок) строят сложные высказывания, т. е. *логические операции* в логике представляют собой действия, образующие новые понятия на основе существующих. Рассмотрим следующие логические операции: конъюнкция, дизъюнкция, инверсия, импликация и эквивалентность.

Отрицанием высказывания A называется высказывание $\neg A$ (читается: «не A » или «неверно, что A »), которое истинно тогда и только тогда, когда высказывание A ложно. Чтобы составить отрицание A достаточно в разговорном языке сказать «неверно, что A ».

Пример

A := «Каспаров – чемпион мира по шахматам».

$\neg A$:= «Неверно, что Каспаров – чемпион мира по шахматам».

Отрицание высказывания A будем обозначать $\neg A$ или \bar{A} . Высказывание \bar{A} истинно, если высказывание A ложно, и ложно, если A истинно. Другими словами, логическое значение высказывания \bar{A} связано с логическим значением высказывания A , как указано в табл. 1, называемой *таблицей истинности операции отрицания*.

Таблица 1

A	\bar{A}
0	1
1	0

Определение отрицания с помощью приведенной таблицы (и других логических операций с помощью таблиц истинности) появилось как результат длительного опыта и полностью оправдало себя на практике.

Конъюнкцией высказываний A и B называется высказывание $A \wedge B$, или $A \& B$ (читается: « A и B »), истинное тогда и только тогда, когда истинны оба высказывания A и B . В разговорной речи конъюнкции соответствует союз «И».

Пример

A := «Треугольник прямоугольный».

B := «Треугольник равнобедренный».

$A \wedge B$:= «Треугольник прямоугольный и равнобедренный».

Логическое значение конъюнкции указано в табл. 2.

Таблица 2

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Практика полностью подтвердила, что именно такое распределение значений истинности наиболее соответствует тому смыслу, который придается в процессе мыслительной деятельности связующему союзу «и».

Дизъюнкцией высказываний A и B называется высказывание $A \vee B$ (читается: « A или B »), ложное тогда и только тогда, когда ложны оба высказывания A и B . В разговорной речи конъюнкции соответствует союз «ИЛИ».

Пример

A := «Петров программист».

B := «Петров филолог».

$A \vee B$:= «Петров программист или филолог».

Логическое значение дизъюнкции указано в табл. 3.

Таблица 3

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Импликацией высказываний A и B называется высказывание $A \rightarrow B$ (читается: « A влечет за собой B » или «из A следует B », или «если A , то B »), ложное тогда и только тогда, когда A истинно, а B ложно.

Пример

$A :=$ «Треугольник равносторонний».

$B :=$ «В треугольнике все углы равны».

$A \rightarrow B :=$ «Если треугольник равносторонний, то все углы равны».

Логическое значение высказывания $A \rightarrow B$ связано с логическим значением высказываний A и B , как указано в табл. 4.

Таблица 4

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

В высказывании $A \rightarrow B$ высказывание A называется посылкой, или антецедентом, а B – следствием, или консеквентом. Импликация призвана отразить процесс рассуждения, умозаключения. Общая характеристика этого процесса следующая: если мы исходим из истинной посылки и правильно (верно) рассуждаем, то мы приходим к истинному заключению (следствию, выводу). Другими словами, если мы исходили из истинной посылки и пришли к ложному выводу, значит, рассуждение построено неверно. Это обосновывает результаты: $1 \rightarrow 0 = 0$ и $1 \rightarrow 1 = 1$. Первые две строки табл. 4 обосновываются практическими примерами.

Импликация играет важную роль в логике высказываний. При учете смыслового содержания высказывания (а не только значений истинности), оборот «если, то» подразумевает причинно-следственную связь. Истинность импликации означает лишь то, что если истинна посылка, то истинно и заключение. При ложной посылке заключение всегда истинно.

Пример

Рассмотрим четыре высказывания:

$A :=$ «Дважды два четыре» = 1;

$B :=$ «Дважды два пять» = 0;

$C := \text{«Снег белый»} = 1;$

$D := \text{«Снег черный»} = 0.$

Образуем четыре импликации:

$A \rightarrow C := \text{«Если дважды два четыре, то снег белый»} = 1 \rightarrow 1 = 1;$

$B \rightarrow C := \text{«Если дважды два пять, то снег белый»} = 0 \rightarrow 1 = 1;$

$A \rightarrow D := \text{«Если дважды два четыре, то снег черный»} = 1 \rightarrow 0 = 0;$

$B \rightarrow D := \text{«Если дважды два пять, то снег черный»} = 0 \rightarrow 0 = 1.$

Эквивалентностью высказываний A и B называется высказывание $A \sim B$, или $A \leftrightarrow B$ (читается: « A эквивалентно B » или « A необходимо и достаточно для B », « A тогда и только тогда, когда B » или « A , если и только если B »), истинное тогда и только тогда, когда оба высказывания A и B одновременно истинны или ложны.

Пример

$A := \text{«Треугольник равнобедренный»}.$

$B := \text{«В треугольнике углы при основании равны»}.$

$A \leftrightarrow B := \text{«Треугольник является равнобедренным тогда и только тогда, когда углы при основании равны»}.$

Логическое значение высказывания $A \rightarrow B$ связано с логическим значением высказываний A и B , как указано в табл. 5.

Таблица 5

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Каждая из введенных логических операций является неким математическим образом, моделью соответствующего союза нашего языка. Эти понятия призваны отразить на языке нулей и единиц соответствующие союзы нашего мышления, которыми человечество пользуется в течение тысячелетий. Так, отрицание, конъюнкция и эквивалентность достаточно точно передают суть логических союзов «не», «и», «тогда и только тогда, когда» соответственно. Сложнее дела обстоят с дизъюнкцией, призванной отразить языковой союз «или». Следует отметить, что кроме рассматриваемой так называемой дизъюнкции в неисключающем смысле (она истинна тогда и только тогда, когда хо-

тя бы один ее член истинен) в некоторых источниках авторами дизъюнкция представляется в исключаящем смысле (строгая дизъюнкция): она истинна тогда и только тогда, когда истинен точно один ее член.

Классификация функций

Число функций от n -переменных $P_2(n)$ определяется как 2^{2^n} .

Логических функций одной переменной $2^{2^1} = 4$, представим их в виде табл. 6.

Таблица 6

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

Функции f_0 и f_3 – константы 0 и 1 соответственно, их значения не зависят от значения переменной, и следовательно, переменная x для них несущественна. Функция f_1 «повторяет» x : $f_1(x) = x$. Функция $f_2(x)$ является отрицанием x . Ее значение противоположно значению x .

Классификация функций двух переменных

Логических функций двух переменных $2^{2^2} = 16$. Рассмотрим классификацию функций двух переменных, приведенную в табл. 7.

Таблица 7

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Функция $f_0(x, y)$ – константа 0, т. е. функция с двумя несущественными переменными.

Функция $f_1(x, y)$ – конъюнкция x и y . Она равна 1, если только x и y равны 1, поэтому ее часто называют функцией «И». Еще одно название – «логическое умножение», поскольку ее таблица действительно совпадает с таблицей обычного умножения для чисел 0 и 1. На естественном языке эта операция выражается соединительными словами: «...и...», «...также...», «как..., так...», «...несмотря на...» и т. п.

Функция $f_2(x, y) = x \wedge \neg y$ – левая импликация.

Функция $f_3(x, y) = x \neg y \vee xy = x$ (y – несущественная переменная).

Функция $f_4(x, y) = \neg x \wedge y$ – *правая компликация*.

Функция $f_5(x, y) = \neg xy \vee xy = y$ (x – несущественная переменная).

Функция $f_6(x, y)$ – *сложение по модулю 2*; ее обозначения: $x \oplus y$, $x \Delta y$. Она равна 1, когда значения ее аргументов различны, и равна 0, когда они равны, поэтому функцию часто называют *неравнозначностью*.

Функция $f_7(x, y)$ – *дизъюнкция* x и y ; ее обозначения: $x \vee y$, $x + y$. Она равна 1, если x или y равны 1 («или» здесь понимается в неразделительном смысле – хотя бы один из двух), поэтому ее часто называют функцией «ИЛИ». В естественном языке эта операция выражается разъединительными словами «...или...», «...либо...» и т. п.

Функция $f_8(x, y) = \neg x \neg y$ – *стрелка Пирса*; ее обозначение $x \downarrow y$.

Функция $f_9(x, y)$ называется *эквивалентностью*, или *равнозначностью*, ее обозначения: $x \sim y$, $x \equiv y$. Она равна 1, когда значения ее аргументов равны, и равна 0, когда они различны: $f_9(x, y) = \neg x \neg y \vee xy = x \equiv y$. На естественном языке это выражается словами «для того чтобы..., необходимо и достаточно...», «...лишь при условии...» и т. п.

Функция $f_{10}(x, y) = \neg x \neg y \vee x \neg y = \neg y$ (x – несущественная переменная).

Функция $f_{11}(x, y) = x \vee \neg y$ – *правая импликация*.

Функция $f_{12}(x, y) = \neg x \neg y \vee \neg xy = \neg x$ (y – несущественная переменная).

Функция $f_{13}(x, y) = \neg x \vee y$ – *импликация (левая импликация)*; ее обозначение $x \rightarrow y$, $x \supset y$; читается: «если x , то y ». На естественном языке эта операция выражается словами «если..., то...», «тогда..., когда...», «постольку..., поскольку...», «при наличии..., следует...», «...есть достаточное условие для...», «...есть необходимое условие для...», «...при условии, что...» и т. п.

Функция $f_{14}(x, y) = \neg x \vee \neg y$ – *итрих Шеффера*, обозначение $x | y$.

Функция $f_{15}(x, y)$ – *константа 1*, т. е. функция с двумя несущественными переменными.

Существенные и несущественные переменные

По определению переменная x_i называется *существенной*, если $f(x_1, \dots, 0, \dots, x_n) \neq f(x_1, \dots, 1, \dots, x_n)$ хотя бы для одного набора значений $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$, где $f = f(x_1, \dots, x_n)$ – булева функция.

При этом наборы $(x_1, \dots, 0, \dots, x_n)$ и $(x_1, \dots, 1, \dots, x_n)$ называются *соседними наборами* по i -й переменной, так как они отличаются только этой координатой.

Переменная x_i в функции $f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)$ называется *несущественной (или фиктивной)*, если $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ при любых значениях остальных переменных, т. е. если изменение значения x_i в любом наборе значений x_1, \dots, x_n не меняет значения функции.

В этом случае функция $f(x_1, \dots, x_n)$ по существу зависит от $n - 1$ переменной, т. е. представляет собой функцию $g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ от $n - 1$ переменной. Говорят, что функция g получена из функции f удалением фиктивной переменной, а функция f получена из g введением фиктивной переменной, причем эти функции по определению считаются равными.

Например, $f(x_1, x_2, x_3) = g(x_1, x_2)$ означает, что при любых значениях x_1 и x_2 $f = g$ независимо от значения x_3 .

Пример

Необходимо доказать, что в функции $f(x_1 \vee \neg x_2)(x_3 \vee \neg x_3)$ переменная x_3 является несущественной.

При $x_3 = 0$: $f(x_1, x_2, 0) \equiv (x_1 \vee \neg x_2)(0 \vee \neg 0) \equiv x_1 \vee \neg x_2 (0 \vee \neg 0 = 1)$.

При $x_3 = 1$: $f(x_1, x_2, 1) \equiv (x_1 \vee \neg x_2)(1 \vee \neg 1) \equiv x_1 \vee \neg x_2 (1 \vee \neg 1 = 1)$.

Так как $f(x_1, x_2, 0) = f(x_1, x_2, 1)$, отсюда следует, что x_3 – несущественная переменная. То есть изменение значения x_3 в любом наборе значений x_1, x_2 не меняет значения функции $f(x_1 \vee \neg x_2)(x_3 \vee \neg x_3)$.

Пример

Необходимо выяснить, какие переменные функции $f(x, y, z) = 01011010$ являются существенными, а какие несущественными и выразить $f(x, y, z)$ формулой, содержащей только существенные переменные.

Представим $f(x, y, z) = 01011010$ в виде табл. 8.

Переменная x существенная, так как, например, наборы $(0, 0, 0)$ и $(1, 0, 0)$ являются соседними по этой переменной и $f(0, 0, 0) \neq f(1, 0, 0)$.

Переменная y является несущественной, так как на всех наборах соседних по данной переменной значения функции равны, т. е. выполняются следующие равенства: $f(0, 0, 0) = f(0, 1, 0)$, $f(1, 0, 0) = f(1, 1, 0)$, $f(0, 0, 1) = f(0, 1, 1)$ и $f(1, 0, 1) = f(1, 1, 1)$.

Таблица 8

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Переменная z является существенной, так как, например, наборы $(0, 0, 0)$ и $(0, 0, 1)$ – соседние по этой переменной и $f(0, 0, 0) \neq f(0, 0, 1)$.

Выпишем таблицу истинности функции $f(x, y, z)$ от существенных переменных – табл. 9.

Таблица 9

x	z	$f(x, y, z)$
0	0	0
0	1	1
1	0	1
1	1	0

Данная функция представляет собой $f(x, y, z) = x \oplus y$.

Представление логической функции

Рассмотрим наиболее распространенные способы задания логической функции.

Формула указывает последовательность логических операций, которые нужно произвести над высказываниями – аргументами, чтобы получить значение функции. Например, $F(X, Y, Z) = X \cdot \bar{Y} \rightarrow Z$.

Таблица истинности. В таблице указываются значения функции в зависимости от значений истинности аргументов. Если функция зависит от n аргументов, то число всех наборов аргументов равно 2^n .

Для построения таблицы истинности выполняется последовательность следующих шагов:

- 1) определение количества переменных n в формуле;
- 2) определение числа строк в таблице $m = 2^n$;
- 3) подсчет количества логических операций в формуле;

4) установление последовательности выполнения логических операций с учетом скобок и приоритетов;

5) определение количества столбцов в таблице: число переменных и число операций;

6) определение наборов входных переменных с учетом того, что они представляют собой натуральный ряд n -разрядных двоичных чисел от 0 до 2^{n-1} ;

7) заполнение таблицы истинности по столбцам, выполняя логические операции в соответствии с установленной на шаге 4 последовательностью.

Наборы входных переменных иногда рекомендуют перечислять следующим образом:

- определить количество наборов входных переменных;
- разделить колонку значений первой переменной пополам и заполнить верхнюю часть колонки 0, а нижнюю – 1;
- разделить колонку значений второй переменной на четыре части и заполнить каждую четверть чередующимися группами 0 или 1, начиная с группы 0;
- продолжать деление колонок значений последующих переменных на 8, 16 и т. д. частей, заполняя их группами 0 или 1 до тех пор, пока группы 0 и 1 не будут состоять из одного символа.

Пример

Необходимо составить таблицу истинности

$$f(x, y) = (x \rightarrow y) \vee (y \rightarrow x).$$

Таблица истинности представлена в табл. 10

Таблица 10

x	y	$(x \rightarrow y)$	$y \rightarrow x$	$f(x, y) = (x \rightarrow y) \vee (y \rightarrow x)$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	1	1	1

Пример

Необходимо составить таблицу истинности

$$f(P, Q, R) = \overline{(P \rightarrow (\overline{\overline{Q \wedge P}})) \rightarrow (P \vee R)}.$$

Таблица 11

P	Q	R	$(Q \wedge P)$	$\overline{(Q \wedge P)}$	$P \rightarrow \overline{(Q \wedge P)}$	$\overline{P \rightarrow \overline{(Q \wedge P)}}$	$(P \vee R)$	$f(P, Q, R)$
0	0	0	0	1	1	0	0	1
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	0	1	1	0	1	1
1	0	0	0	1	0	1	1	1
1	0	1	0	1	0	1	1	1
1	1	0	1	0	1	0	1	1
1	1	1	1	0	1	0	1	1

Таблица истинности представлена в табл. 11.

Числовой способ задания функции. Каждой независимой переменной-аргументу функции ставится в соответствие число 2^k ($k = 0, 1, 2, \dots$). Аргументы функции записываются в виде упорядоченного множества, например, $F(X_1, X_2, X_3)$. При этом переменная, записанная крайней справа, получает коэффициент $2^0 = 1$; переменная, стоящая рядом слева, – коэффициент $2^1 = 2$ и т. д.

Так, для функции $F(X_1, X_2, X_3)$ независимые переменные получают следующие коэффициенты: $X_3 - 1$, $X_2 - 2$, $X_1 - 4$. Для каждого набора независимых переменных определяется номер N по формуле

$$N = 4 \cdot X_1 + 2 \cdot X_2 + 1 \cdot X_3.$$

При задании функции указывают номера тех наборов, на которых функция равна единице, перед списком номеров единичных наборов ставят знак дизъюнкции. Можно также указать те номера наборов, на которых функция равна нулю, но при этом перед списком нулевых наборов ставят знак конъюнкции.

Пример

Функция, заданная таблицей истинности, представлена в табл. 12.

Таблица 12

N	X	Y	Z	$F(X, Y, Z)$
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0

Окончание табл. 12

N	X	Y	Z	$F(X, Y, Z)$
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Числовым способом задания функция может быть представлена следующим образом: $F(X, Y, Z) = \vee(0, 1, 4, 7) = \wedge(2, 3, 5, 6)$.

Геометрический способ задания логической функции. Для функции n – независимых логических переменных – рассматривается единичный n -мерный куб. Вершины куба соответствуют наборам независимых переменных. Каждой вершине приписывают значение функции на соответствующем наборе. На рисунке единичные наборы помечают, например, кружками.

Геометрический способ задания логической функции приведен на рис. 1.

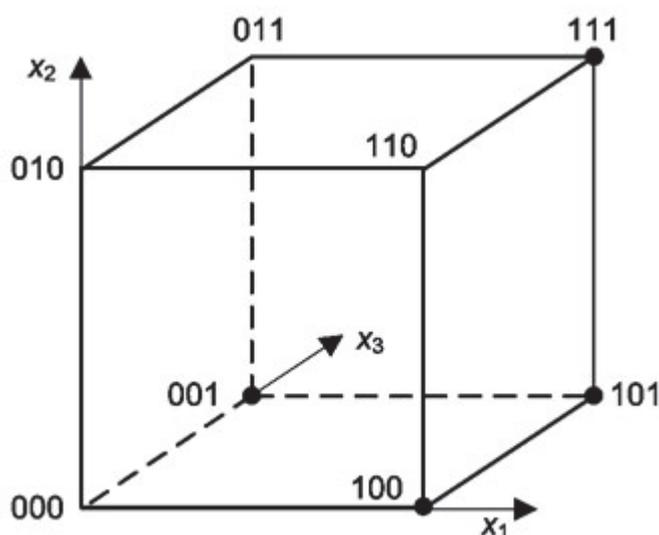


Рис. 1. Геометрический способ задания логической функции

Логическая схема представляет собой условное графическое обозначение логических функций. Примером логической схемы является релейно-контактная схема.

Под *релейно-контактной схемой* (РКС) понимается устройство из проводников и двухпозиционных контактов. Контакты могут быть

двух видов: *замыкающие* и *размыкающие*. Каждый контакт подключен к некоторому реле (переключателю). К одному реле может быть подключено несколько контактов – как замыкающих, так и размыкающих. Когда через реле проходит электрический ток, то все подключенные к нему замыкающие контакты замыкаются, а размыкающие – размыкаются. При отсутствии тока в реле все подключенные к нему замыкающие контакты разомкнуты, а размыкающие – замкнуты.

Каждому реле ставится в соответствие своя булева переменная x_1, x_2, \dots, x_n , которая принимает значение 1, если через реле проходит электрический ток, и значение 0 – если реле отключено.

На схеме все замыкающие контакты, подключенные к реле x , обозначаются символом x ; размыкающие контакты, подключенные к этому реле, – отрицанием \bar{x} .

Таким образом, если через реле x проходит ток, то всем его замыкающим контактам x соответствует значение 1, а размыкающим контактам \bar{x} – 0. При отключенном реле x замыкающим контактам x соответствует значение 0, а размыкающим – 1.

Всей релейно-контактной схеме ставится в соответствие булева функция f , зависящая от переменных x_1, x_2, \dots, x_n . Если при данном наборе состояний реле x_1, x_2, \dots, x_n (через некоторые из которых проходит ток, а некоторые отключены) схема проводит ток, то переменная f принимает значение 1. В противном случае переменная f принимает значение 0.

Таким образом, релейно-контактная схема с n независимыми реле x_1, x_2, \dots, x_n определяет булеву функцию $f(x_1, x_2, \dots, x_n)$, которая называется *функцией проводимости релейно-контактной схемы*.

Пример

Релейно-контактная схема с последовательным соединением контактов представлена на рис. 2.



Рис. 2. Последовательное соединение контактов релейно-контактной схемы

Функцией проводимости данной релейно-контактной схемы является конъюнкция $f(x, y) = x \wedge y$.

Пример

Релейно-контактная схема с параллельным соединением контактов представлена на рис. 3.

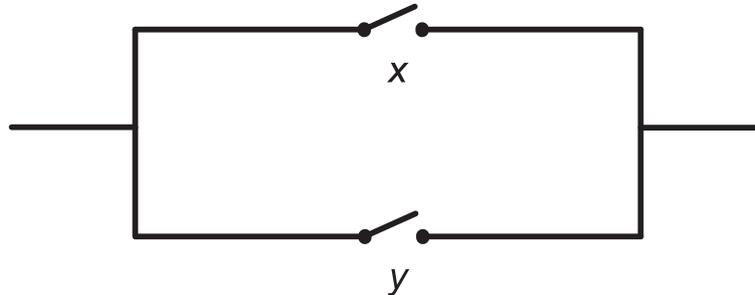


Рис. 3. Параллельное соединение контактов релейно-контактной схемы

Функцией проводимости данной релейно-контактной схемы является дизъюнкция $f(x, y) = x \vee y$.

Всякая булева функция может быть реализована с помощью релейно-контактной схемы, т. е. может быть построена такая схема, для которой данная булева функция будет функцией проводимости.

Пример

Построить релейно-контактную схему с заданной функцией проводимости $f(x, y, z) = (\bar{x} \wedge (\bar{y} \wedge z \vee x \vee y))$.

Схема представляет собой последовательное соединение контакта \bar{x} и схемы с функцией проводимости $(\bar{y} \wedge z \vee x \vee y)$. Последняя схема, в свою очередь, состоит из трех параллельных ветвей. В первой ветви последовательно соединены контакты \bar{y} и z , вторая включает x , третья – контакт y .

Таким образом, релейно-контактная схема имеет вид, представленный на рис. 4.

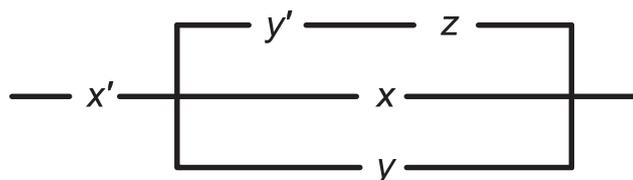


Рис. 4. Релейно-контактная схема с функцией проводимости $f(x, y, z) = (\bar{x} \wedge (\bar{y} \wedge z \vee x \vee y))$

Пример

Построить релейно-контактную схему с заданной функцией проводимости $f(x, y, z, t) = ((\bar{x} \vee y) \vee (\bar{z} \wedge t)) \wedge (x \vee \bar{y} \vee z)$.

Схема представляет собой последовательное соединение схемы с функцией проводимости $(\bar{x} \vee y) \vee (\bar{z} \wedge t)$ и схемы с функцией проводимости $(x \vee \bar{y} \vee z)$.

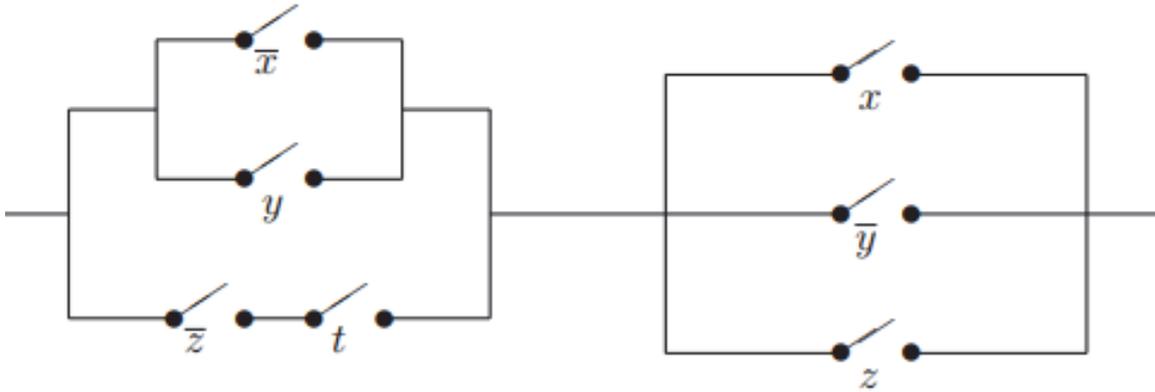


Рис. 5. Релейно-контактная схема с функцией проводимости

$$f(x, y, z, t) = ((\bar{x} \vee y) \vee (\bar{z} \wedge t)) \wedge (x \vee \bar{y} \vee z)$$

Схема с функцией проводимости $(\bar{x} \vee y) \vee (\bar{z} \wedge t)$ состоит из двух параллельных ветвей. Первая ветвь, в свою очередь, также состоит из двух ветвей \bar{x} и y , вторая представляет собой последовательное соединение \bar{z} и t . Схема с функцией проводимости $(x \vee \bar{y} \vee z)$, в свою очередь, состоит из трех параллельных ветвей. Первая ветвь включает контакт x , вторая – \bar{y} , третья – контакт z (рис. 5).

1.2. Формулы алгебры логики

Понятие формулы алгебры логики

Всякое сложное высказывание, которое может быть получено из элементарных высказываний посредством применения логических связей отрицания, конъюнкции, дизъюнкции, импликации и эквивалентности, называют *формулой алгебры логики*.

Символы x_1, \dots, x_n будем считать формулами глубины 0.

Формула $F = f(F_1, \dots, F_n)$ имеет глубину $k + 1$, если f – функция, а F_1, \dots, F_n – формулы, максимальная из глубин которых равна k . При

этом F_1, \dots, F_n называются *подформулами* F , а f – *внешней или главной операцией* F .

Рассмотрим более точное определение *формулы алгебры логики*.

1. Любую пропозициональную переменную можно назвать формулой нулевого порядка, т. е. $A_i = F_i$.

2. Если F_1 и F_2 – пропозициональные формулы, то $\neg F_1$; $\neg F_2$; $F_1 \wedge F_2$; $F_1 \vee F_2$; $F_1 \rightarrow F_2$ и $F_1 \leftrightarrow F_2$ – также пропозициональные формулы.

3. Никаких других формул в исчислении высказываний нет.

Множество формул образует язык математической логики. Это множество перечислимо и разрешимо. *Подформула формулы* – любая ее часть, которая сама является формулой.

Для формирования сложных формул используют вспомогательные символы «(» и «)».

При записи сложных формул следует помнить, что:

- каждое вхождение логической связки « \neg » относится к пропозициональной переменной или формуле, следующей непосредственно за логической связкой справа;

- каждое вхождение логической связки « \wedge » после расстановки скобок связывает пропозициональные переменные или формулы, непосредственно окружающие логическую связку;

- каждое вхождение логической связки « \vee » после расстановки скобок связывает пропозициональные переменные или формулы, непосредственно окружающие эту связку, и т. д.

При использовании этих правил к одной и той же формуле скобки следует расставлять постепенно, продвигаясь слева направо.

Приоритет выполнения логических связок (порядок логических связок по силе и значимости) следующий: \neg ; \wedge ; \vee ; \rightarrow ; \leftrightarrow . То есть самой сильной связкой является отрицание, затем конъюнкция, дизъюнкция, импликация и, наконец, эквивалентность. Учитывая приоритет логических связок, возможно опускать некоторые пары скобок.

Для каждой формулы должна существовать конечная последовательность всех ее подформул, т. е. такая конечная последовательность, которая начинается с входящих в данную формулу пропозициональных переменных, заканчивается самой этой формулой, и каждый член этой последовательности, не являющийся пропозициональной переменной, есть либо отрицание уже имеющегося члена этой последовательности, либо получается из двух уже имеющихся членов этой по-

следовательности их соединением с помощью одного из знаков \wedge , \vee , \rightarrow , \leftrightarrow и заключением полученного выражения в скобки.

Такую последовательность всех подформул данной формулы иногда называют *порождающей последовательностью* для данной формулы. Наличие такой последовательности у логического выражения служит критерием того, что выражение является формулой. Это свойство отличает формулы.

Пример

Необходимо удалить лишние скобки в формуле

$$F = (((F_1 \vee (\neg F_2)) \rightarrow F_3) \leftrightarrow F_4).$$

Решение:

1) убрать внешние скобки для формулы, так как они не определяют приоритет операций:

$$F = ((F_1 \vee (\neg F_2)) \rightarrow F_3) \leftrightarrow F_4;$$

2) убрать скобки, охватывающие формулу \rightarrow , так как операция \leftrightarrow будет выполняться после \rightarrow :

$$F = (F_1 \vee (\neg F_2)) \rightarrow F_3 \leftrightarrow F_4;$$

3) убрать скобки, охватывающие формулу \vee , так как операция \rightarrow будет выполняться после \vee :

$$F = F_1 \vee (\neg F_2) \rightarrow F_3 \leftrightarrow F_4;$$

4) убрать скобки, охватывающие формулу отрицания, так как операция \vee будет исполняться только после выполнения операции отрицания:

$$F = F_1 \vee \neg F_2 \rightarrow F_3 \leftrightarrow F_4.$$

Итак, последовательность исполнения операций после задания значений пропозициональных переменных следующая: сначала необходимо определить значение формулы $(\neg F_2)$, затем $(F_1 \vee (\neg F_2))$, затем $((F_1 \vee (\neg F_2)) \rightarrow F_3)$ и, наконец, $((F_1 \vee (\neg F_2)) \rightarrow F_3) \leftrightarrow F_4$.

Пример

Необходимо расставить скобки в формуле

$$F = F_1 \wedge F_2 \wedge F_3 \vee \neg F_1 \rightarrow F_3 \leftrightarrow F_1.$$

1) поставить скобки на формулу, реализующую операцию отрицания:

$$F = F_1 \wedge F_2 \wedge F_3 \vee (\neg F_1) \rightarrow F_3 \leftrightarrow F_1;$$

2) поставить скобки на формулу, реализующую операцию конъюнкции:

$$F = ((F_1 \wedge F_2) \wedge F_3) \vee (\neg F_1) \rightarrow F_3 \leftrightarrow F_1;$$

3) поставить скобки на формулу, реализующую операцию дизъюнкции:

$$F = (((F_1 \wedge F_2) \wedge F_3) \vee (\neg F_1)) \rightarrow F_3 \leftrightarrow F_1;$$

4) поставить скобки на формулу, реализующую операцию импликации:

$$F = (((((F_1 \wedge F_2) \wedge F_3) \vee (\neg F_1))) \rightarrow F_3 \leftrightarrow F_1;$$

5) поставить скобки на формулу, реализующую операцию эквивалентности:

$$F = ((((((F_1 \wedge F_2) \wedge F_3) \vee (\neg F_1))) \rightarrow F_3 \leftrightarrow F_1).$$

Логическое следствие

Одной из основных задач логики является нахождение общих способов установления связей логических значений одних высказываний с логическими значениями других высказываний на основе исследования формальной структуры высказываний. Это позволяет установить структуру высказываний и выявить закономерности, что из чего следует. С помощью таких знаний происходит доказательство математических теорем и, следовательно, развитие математики. В повседневной жизни это служит инструментом рассуждений, обоснований и доказательств.

Теория логического следования в рамках алгебры логики изучает закономерности образования формул F_1, F_2, \dots, F_m, H , по которым первые m формул связаны с формулой H отношением логического следствия.

Рассмотрим формальное определение понятия логического следования. Формула алгебры логики $H(X_1, X_2, \dots, X_n)$ называется *логическим следствием* формул $F_1(X_1, X_2, \dots, X_n), \dots, F_m(X_1, X_2, \dots, X_n)$, если формула $H(X_1, X_2, \dots, X_n)$ превращается в истинное высказывание при всякой такой подстановке вместо всех ее пропозициональных пере-

менных X_1, X_2, \dots, X_n конкретных высказываний, при которой в истинное высказывание превращаются все формулы $F_1(X_1, X_2, \dots, X_n), \dots, F_m(X_1, X_2, \dots, X_n)$.

Таким образом, $F_1, \dots, F_m \models H$, если для любого набора высказываний X_1, X_2, \dots, X_n из равенств $F_1(X_1, X_2, \dots, X_n) = 1, \dots, F_m(X_1, X_2, \dots, X_n) = 1$ следует равенство

$$H(X_1, X_2, \dots, X_n) = 1.$$

Запись $F_1, \dots, F_m \models H$ обозначает, что формула H является логическим следствием формул F_1, F_2, \dots, F_m . При этом формулы F_1, F_2, \dots, F_m называются *посылками* для логического следствия H . Также логическое следствие может быть обозначено как $\frac{F_1, F_2, \dots, F_m}{H}$.

Алгоритм проверки формул на логическое следствие просматривает последовательно по строкам таблицы значения формул F_1, \dots, F_m, H .

Если хотя бы один элемент F_1, \dots, F_m нулевой строки равен 0, то без просмотра значения формулы H в этой строке происходит переход к просмотру следующей строки.

Если все элементы F_1, \dots, F_m нулевой строки равны 1, то просматривается значение формулы H в этой строке: если значение формулы H в строке равно 0, выдается результат, что формула H не является логическим следствием формул F_1, \dots, F_m ; если оно равно 1, то осуществляется переход к просмотру следующей строки и т. д.

Если после просмотра последней строки должен произойти переход к просмотру следующей строки, то это означает, что определение логического следствия выполнено и формула H является логическим следствием формул F_1, \dots, F_m .

Пример

Необходимо проверить логическое следование: если $F_1(X, Y) = X \wedge Y$, $H(X, Y) = X \vee Y$, то $F_1 \models H$. Для проверки построим таблицу истинности (табл. 13).

Таблица 13

N	X	Y	$F_1(X, Y) = X \wedge Y$	$H(X, Y) = X \vee Y$
0	0	0	0	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	1

Сравним построчно значения столбцов таблицы истинности $F_1(X, Y)$ и $H(X, Y)$. В строке $N = 0$ столбца $F_1(X, Y)$ стоит 0, следовательно, осуществляем переход к строке $N = 1$. Во строке $N = 1$ столбца $F_1(X, Y)$ стоит 0, переходим к строке $N = 2$. В строке $N = 2$ столбца $F_1(X, Y)$ стоит 1, необходимо проверить, стоит ли 1 в строке столбца $H(X, Y)$. Так как это действительно так, то можно осуществлять переход к строке $N = 3$. Аналогично просматриваем строку $N = 3$.

Таким образом, произведено сопоставление логических значений формул $F_1(X, Y)$ и $H(X, Y)$ при всевозможных наборах значений их пропозициональных переменных и определено, что формула $H(X, Y)$ является логическим следствием формулы $F_1(X, Y)$.

Пример

Необходимо проверить логическое следование: если

$$F_1(X, Y, Z) = X \wedge Y, F_2(X, Y, Z) = Z \rightarrow X, H(X, Y, Z) = X \rightarrow (Y \vee Z),$$

то $F_1, F_2 \models H$. Для проверки построим таблицу истинности (табл. 14).

Таблица 14

X	Y	Z	$F_1(X, Y, Z) = X \wedge Y$	$F_2(X, Y, Z) = Z \rightarrow X$	$H(X, Y, Z) = X \rightarrow (Y \vee Z)$
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Сравним построчно значения столбцов таблицы истинности $F_1(X, Y, Z)$, $F_2(X, Y, Z)$ и $H(X, Y, Z)$. Последовательный просмотр по строкам столбцов $F_1(X, Y, Z)$, $F_2(X, Y, Z)$ и $H(X, Y, Z)$ показывает, что как только в какой-либо строке столбцов $F_1(X, Y, Z)$, $F_2(X, Y, Z)$ все элементы равны 1, то в столбце $H(X, Y, Z)$ также значение 1. Значит, требуемое логическое следование действительно выполняется.

Эквивалентные формулы

Формулы $F = (X_1, X_2, \dots, X_n)$ и $G = (X_1, X_2, \dots, X_n)$ алгебры логики называются *равносильными*, или *эквивалентными*, если они при-

нимают одинаковые логические значения на любом наборе значений входящих в формулы пропозициональных переменных, т. е. для любой интерпретации φ выполняется равенство $\varphi(F) = \varphi(G)$.

Пример

Покажем, что формулы $F = \bar{x}$ и $G = (\bar{y} \wedge (y \vee \bar{x})) \vee \bar{x}$ эквивалентны. Для этого построим таблицу истинности и убедимся, что логические значения формул на любом наборе значений входящих в формулы пропозициональных переменных совпадают (табл. 15).

Таблица 15

X	Y	$F = \bar{x}$	$(y \vee \bar{x})$	$(\bar{y} \wedge (y \vee \bar{x}))$	$G = (\bar{y} \wedge (y \vee \bar{x})) \vee \bar{x}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	0	0	0	0
1	1	0	1	0	1

Пример

Покажем, что формулы $F = \bar{x}$ и $G = \bar{y} \wedge (y \vee \bar{x})$ не эквивалентны. Для этого построим таблицу истинности и убедимся, что существуют наборы, на которых логические значения формул не совпадают (табл. 16).

Таблица 16

X	Y	$F = \bar{x}$	$(y \vee \bar{x})$	$G = \bar{y} \wedge (y \vee \bar{x})$
0	0	1	1	1
0	1	1	1	0
1	0	0	0	0
1	1	0	1	0

Классы логических формул

Формула F называется *тождественно истинной (тавтологией)*, если для любой интерпретации φ выполняется равенство $\varphi(F) = 1$.

Например, формула $F = X \wedge Y \rightarrow X$ является тождественно истинной. Таблица истинности формулы F – табл. 17.

Столбец формулы F состоит из одних единиц. Это означает, что формула F тождественно истинна (тавтология).

Таблица 17

X	Y	$X \wedge Y$	$F = X \wedge Y \rightarrow X$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Например, формула $F = X \leftrightarrow (X \wedge Y)$ не является тождественно истинной. Таблица истинности формулы F – табл. 18.

Таблица 18

X	Y	$X \wedge Y$	$F = X \leftrightarrow (X \wedge Y)$
0	0	0	1
0	1	0	1
1	0	0	0
1	1	1	1

Так как существует интерпретация $F(1, 0) = 0$, то формула F не является тождественно истинной (тавтологией).

Формула F называется *тождественно ложной* (противоречивой), если для любой интерпретации φ выполняется равенство $\varphi(F) = 0$.

Пример

Докажем тождественную ложность формулы $F = X \wedge (\overline{Y \rightarrow X})$ методом подстановки значений переменных:

$$F(0, 0) = 0 \wedge (\overline{0 \rightarrow 0}) = 0;$$

$$F(0, 1) = 0 \wedge (\overline{1 \rightarrow 0}) = 0;$$

$$F(1, 0) = 1 \wedge (\overline{0 \rightarrow 1}) = 0;$$

$$F(1, 1) = 1 \wedge (\overline{1 \rightarrow 1}) = 0.$$

При любых значениях пропозициональных переменных формула равна 0. Это означает, что формула F тождественно ложна (противоречива).

Формула F называется *выполнимой* (опровержимой), если существует интерпретация, при которой формула F истинна (ложна). Эта терминология применима также к множествам формул: множество Q

формул *выполнимо*, если существует интерпретация, при которой истинны все формулы Q .

Пример

Формула $F = \bar{X} \rightarrow (Y \rightarrow X)$ выполнима, так как, например $F(0,0) = \bar{0} \rightarrow (0 \rightarrow 0) = 1 \rightarrow 1 = 1$. Но данная формула не является тавтологией, так как $F(0,1) = \bar{0} \rightarrow (1 \rightarrow 0) = 1 \rightarrow 0 = 0$.

Пример

Формула $F = \bar{X} \rightarrow (Y \rightarrow X)$ выполнима, так как существуют наборы, на которых она равна 0, но данная формула не является тавтологией, так как на наборе (0, 1) она равна 1 (табл. 19).

Таблица 19

X	Y	$F = \bar{X} \rightarrow (Y \rightarrow X)$
0	0	1
0	1	0
1	0	1
1	1	1

Признак равносильности формул: две формулы F и G равносильны тогда и только тогда, когда формула $F \leftrightarrow G$ является тавтологией.

Пример

Формулы $F = X \rightarrow Y$ и $F = \bar{Y} \rightarrow \bar{X}$ равносильны, так как формула $X \rightarrow Y \leftrightarrow \bar{Y} \rightarrow \bar{X}$ является тавтологией (табл. 20).

Таблица 20

X	Y	$X \rightarrow Y$	$\bar{Y} \rightarrow \bar{X}$	$X \rightarrow Y \leftrightarrow \bar{Y} \rightarrow \bar{X}$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	1
1	1	1	1	1

1.3. Законы алгебры логики

Булевой алгеброй, или алгеброй логики, называется множество всех логических функций с *булевыми операциями*: дизъюнкции,

конъюнкции и отрицания. Булевы операции подчиняются следующим законам.

1. Ассоциативный закон:

а) $x(yz) = (xy)z$;

б) $x \vee (y \vee z) = (x \vee y) \vee z$.

2. Коммутативный закон:

а) $xy = yx$;

б) $x \vee y = y \vee x$.

3. Дистрибутивный (распределительный) закон:

а) $x(y \vee z) = xy \vee xz$;

б) $x \vee (yz) = (x \vee y)(x \vee z)$.

4. Закон идемпотентности:

а) $xx = x$;

б) $x \vee x = x$.

5. Двойное отрицание:

$\overline{\overline{x}} = x$.

6. Свойства констант:

а) $x \wedge 1 = x$; б) $x \wedge 0 = 0$;

в) $x \vee 1 = 1$; г) $x \vee 0 = x$;

д) $\overline{1} = 0$; е) $\overline{0} = 1$.

7. Закон противоречия:

$x\overline{x} = 0$.

8. Закон исключения третьего:

$x \vee \overline{x} = 1$.

9. Законы де Моргана:

а) $\overline{xy} = \overline{x} \vee \overline{y}$;

б) $\overline{x \vee y} = \overline{x} \overline{y}$.

В справедливости законов алгебры логики можно убедиться подстановкой в обе части всех возможных наборов значений аргументов. Метод подстановок типичен для алгебры логики. Эквивалентные соотношения законов алгебры логики остаются справедливыми при подстановке вместо переменных x любых логических функций F , если при этом соблюдено *правило подстановки формулы вместо переменной*: при подстановке формулы F вместо переменной x все вхождения переменной x в исходное выражение надо одновременно заменить формулой F .

Правила упрощения булевых функций

Знание законов алгебры высказываний позволяет выполнять эквивалентные преобразования любых логических формул, т. е. такие преобразования, которые сохраняют значения логических функций для любых наборов независимых переменных. Они применяются для упрощения логических функций.

1. Правило поглощения:

$$а) x \vee xz = x;$$

$$б) x(x \vee y) = x.$$

2. Правило склеивания:

$$xy \vee x\bar{y} = x.$$

3. Правило раскрепления (обратное к склеиванию):

$$x = xy \vee x\bar{y}.$$

4. Обобщенное склеивание:

$$xz \vee y\bar{z} \vee xy = xz \vee y\bar{z}.$$

5. Удаление отрицания:

$$x \vee \bar{x}y = x \vee y.$$

Преобразования основных логических операций к булевым операциям осуществляются следующим образом:

$$x \rightarrow y \equiv \bar{x} \vee y \equiv \overline{\bar{x}y};$$

$$x \leftrightarrow y \equiv (x \rightarrow y)(y \rightarrow x) \equiv (\bar{x} \vee y)(\bar{y} \vee x) \equiv \overline{\bar{x}y} \vee \overline{\bar{y}x};$$

$$x \oplus y \equiv (\bar{x}y \vee x\bar{y}); \quad x \downarrow y \equiv \overline{\bar{x}y}; \quad x|y \equiv \bar{x} \vee \bar{y}.$$

Алгоритм упрощения булевой функции

1. Перейти в систему булевых операций (конъюнкция, дизъюнкция и отрицание).

2. С помощью закона де Моргана опустить все отрицания до элементарных переменных и удалить двойные отрицания по закону двойного отрицания.

3. Применить законы идемпотентности и противоречия, сократить лишние конъюнкции (дизъюнкции) и повторения переменных в конъюнкциях (дизъюнкциях).

4. Упростить выражение, удалив все константы, применяя свойства констант.

Пример

Необходимо упростить выражение

$$F = (F_1 \rightarrow F_2) \rightarrow ((F_2 \rightarrow F_3) \rightarrow (F_1 \vee F_2 \rightarrow F_3)).$$

1. Удалить всюду логическую связку « \rightarrow » (перейти к булевым операциям):

$$F = (\overline{\overline{F_1} \vee F_2}) \vee ((\overline{\overline{F_2} \vee F_3}) \vee (\overline{\overline{F_1} \vee F_2} \vee F_3)).$$

2. Опустить отрицание на элементарные формулы по закону де Моргана и удалить двойные отрицания по закону двойного отрицания:

$$F = (F_1 \wedge \overline{F_2}) \vee (F_2 \wedge \overline{F_3}) \vee (\overline{F_1} \wedge \overline{F_2}) \vee F_3.$$

3. Выполнить преобразование по закону дистрибутивности:

$$F = ((F_1 \vee \overline{F_1}) \wedge \overline{F_2}) \vee (F_2 \wedge \overline{F_3}) \vee F_3.$$

4. По закону исключения третьего удалить элемент $(F_1 \vee \overline{F_1})$:

$$F = \overline{F_2} \vee (F_2 \wedge \overline{F_3}) \vee F_3.$$

5. Выполнить преобразование по закону дистрибутивности:

$$F = \overline{F_2} \vee (F_2 \vee F_3) \wedge (\overline{F_3} \vee F_3).$$

6. По закону исключения третьего удалить элемент $(\overline{F_3} \vee F_3)$:

$$F = \overline{F_2} \vee (F_2 \vee F_3).$$

7. Применить закон ассоциативности: $F = (\overline{F_2} \vee F_2) \vee F_3$.

8. Применить закон исключения третьего: $F = 1 \vee F_3$.

9. Приравнять «истине» значение формулы F согласно закону преобразования выражений с константами.

Пример

Необходимо упростить выражение $F = (x \rightarrow y)(y \rightarrow \overline{x})(z \rightarrow x)$.

Решение:

1. Удалить всюду логическую связку « \rightarrow » (перейти к булевым операциям):

$$F = (\overline{x} \vee y)(\overline{y} \vee \overline{x})(\overline{z} \vee x).$$

2. Выполнить преобразование по закону дистрибутивности:

$$\begin{aligned} F &= ((\overline{x} \vee y)\overline{y}) \vee ((\overline{x} \vee y)\overline{x})(\overline{z} \vee x) \equiv (\overline{x}\overline{y} \vee y\overline{y}) \vee (\overline{x}\overline{x} \vee y\overline{x})(\overline{z} \vee x) \equiv \\ &\equiv \overline{x}\overline{y} \vee y\overline{y} \vee \overline{x}\overline{x}z \vee \overline{x}\overline{x}x \vee y\overline{x}z \vee y\overline{x}x. \end{aligned}$$

3. По закону идемпотентности исключить повторяющиеся элементы:

$$F = \overline{\overline{xy}} \vee \overline{y\overline{y}} \vee \overline{\overline{xz}} \vee \overline{\overline{xx}} \vee \overline{yxz} \vee \overline{yxx}.$$

4. По закону противоречия удалить элементы $\overline{y\overline{y}}$ и $\overline{\overline{xx}}$:

$$F = \overline{\overline{xy}} \vee \overline{\overline{xz}} \vee \overline{yxz}.$$

5. Применить правило поглощения $\overline{\overline{xz}} \vee \overline{yxz} \equiv \overline{\overline{xz}}$: $F = \overline{\overline{xy}} \vee \overline{\overline{xz}}$.

6. Применить закон дистрибутивности и получить упрощенную формулу: $F = \overline{x}(\overline{y} \vee \overline{z})$.

Пример

Необходимо упростить выражение

$$F = F_1 \wedge (F_1 \vee F_2) \vee F_2 \wedge (F_1 \vee \overline{F_2}).$$

1) по закону дистрибутивности:

$$F = F_1 \wedge F_1 \vee F_1 \wedge F_2 \vee F_1 \wedge F_2 \vee F_2 \wedge \overline{F_2}.$$

2) по законам идемпотентности и противоречия: $F = F_1 \vee F_1 \wedge F_2$.

3) по закону поглощения: $F = F_1$.

1.4. Стандартные формы представления формул алгебры логики

Нормальные формы формул алгебры логики

Дизъюнктивная нормальная форма (ДНФ) формулы есть формула, равносильная формуле исходной логической функции и записанная в виде дизъюнкции элементарных конъюнкций, построенных на пропозициональных переменных, т. е.

$$F = K_1 \vee K_2 \vee K_3 \vee \dots,$$

где $K_i = (A \wedge B \wedge C \wedge \dots)$.

Элементарной конъюнкцией, или *конъюнктивным одночленом* от переменных A, B, C, \dots называется конъюнкция каких-либо из этих переменных или их отрицаний.

В элементарной конъюнкции нет двух одинаковых пропозициональных переменных, так как по закону идемпотентности $F \wedge F = F$.

В ДНФ нет двух одинаковых элементарных конъюнкций, так как по закону идемпотентности $F \vee F = F$.

Если одна из элементарных конъюнкций содержит F и \overline{F} , то элементарную конъюнкцию следует удалить, так как $F \wedge \overline{F} = 0$.

Конъюнктивная нормальная форма (КНФ) формулы есть формула, равносильная формуле исходной логической функции и записанная в виде конъюнкции элементарных дизъюнкций, построенных на пропозициональных переменных, т. е.

$$F = D_1 \wedge D_2 \wedge D_3 \wedge \dots,$$

где $D_i = (A \vee B \vee C \vee \dots)$.

Элементарной дизъюнкцией, или *дизъюнктивным одночленом* от переменных A, B, C, \dots называется дизъюнкция каких-либо из этих переменных или их отрицаний.

В элементарной дизъюнкции нет двух одинаковых пропозициональных переменных, так как по закону идемпотентности $F \vee F = F$.

В КНФ нет двух одинаковых элементарных дизъюнкций, так как по закону идемпотентности $F \wedge F = F$.

Если одна из элементарных дизъюнкций содержит F и \overline{F} , то следует удалить, так как $F \vee \overline{F} = 1$.

Наибольшее распространение в логике высказываний получили формулы вида КНФ, элементарные дизъюнкции которых D_i принято называть *дизъюнктами*, а члены каждого дизъюнкта A, B, C – *атомами*.

Алгоритм приведения к нормальной форме

Шаг 1. Устранить логические связи « \leftrightarrow » и « \rightarrow » всюду по правилам:

$$F_1 \leftrightarrow F_2 \equiv (F_1 \rightarrow F_2)(F_2 \rightarrow F_1) \equiv (\overline{F_1} \vee F_2)(\overline{F_2} \vee F_1) \equiv \overline{F_1}F_2 \vee F_1F_2;$$

$$F_1 \rightarrow F_2 \equiv \overline{F_1} \vee F_2 \equiv (\overline{F_1 F_2}).$$

Шаг 2. Продвинуть отрицание до элементарной формулы (пропозициональной переменной) по законам де Моргана и двойного отрицания.

Шаг 3. Применить закон дистрибутивности:

$$\text{для КНФ} - F_1 \vee (F_2 F_3) \equiv (F_1 \vee F_2)(F_1 \vee F_3);$$

$$\text{для ДНФ} - F_1(F_2 \vee F_3) \equiv F_1 F_2 \vee F_1 F_3.$$

Пример

Дана формула $F = ((A \rightarrow B) \rightarrow (C \rightarrow A)) \rightarrow (\bar{B} \rightarrow \bar{C})$. Необходимо привести формулу к виду КНФ и ДНФ.

1. Устранить \rightarrow :

$$F = \overline{\overline{(\bar{A} \vee B)} \vee \overline{(C \vee A)}} \vee (\bar{B} \vee \bar{C}).$$

2. Продвинуть отрицание до элементарной формулы:

$$F = ((\bar{A} \vee B) \wedge (C \wedge \bar{A})) \vee (B \vee \bar{C}).$$

3. Применить закон дистрибутивности:

$$F = (\bar{A} \vee B \vee B \vee \bar{C}) \wedge (C \wedge \bar{A} \vee B \vee \bar{C}).$$

4. Применить закон идемпотентности:

$$F = (\bar{A} \vee B \vee \bar{C}) \wedge (C \wedge \bar{A} \vee B \vee \bar{C}).$$

5. Применить закон дистрибутивности:

$$F = (\bar{A} \vee B \vee \bar{C}) \wedge (C \vee B \vee \bar{C}) \wedge (\bar{A} \vee B \vee \bar{C}).$$

6. Применить закон идемпотентности:

$$F = (\bar{A} \vee B \vee \bar{C}) \wedge (C \vee B \vee \bar{C}).$$

7. Применить закон исключения третьего:

$$F = (\bar{A} \vee B \vee \bar{C}) \wedge (B \vee 1).$$

8. Удалить константы по свойствам констант: $F = (\bar{A} \vee B \vee \bar{C})$.

Данная формула одновременно является ДНФ и КНФ формулы.

Пример

Дана формула $F = ((F_1 \rightarrow (F_2 \vee \bar{F}_3)) \rightarrow F_4)$. Необходимо привести формулу к виду КНФ.

1. Избавиться от импликации: $F = (\bar{F}_1 \vee (F_2 \vee \bar{F}_3)) \rightarrow F_4$.

2. Избавиться от импликации: $F = \overline{\bar{F}_1 \vee (F_2 \vee \bar{F}_3)} \vee F_4$.

3. Применить законы де Моргана и двойного отрицания:

$$F = F_1 \wedge \bar{F}_2 \wedge F_3 \vee F_4.$$

4. Применить закон дистрибутивности:

$$F = (F_1 \vee F_4) \wedge (\overline{F_2} \vee F_4) \wedge (F_3 \vee F_4).$$

Пример

Дана формула $F = ((\overline{F_1} \wedge \overline{F_2}) \wedge (F_1 \vee F_2))$. Необходимо привести формулу к виду ДНФ.

1. Применить закон де Моргана: $F = (\overline{F_1} \vee \overline{F_2}) \wedge (F_1 \vee F_2)$.

2. Применить закон дистрибутивности:

$$F = ((\overline{F_1} \vee \overline{F_2}) \wedge F_1) \vee ((\overline{F_1} \vee \overline{F_2}) \wedge F_2).$$

3. Применить закон дистрибутивности:

$$F = (\overline{F_1} \wedge F_1) \vee (\overline{F_2} \wedge F_1) \vee (\overline{F_1} \wedge F_2) \vee (\overline{F_2} \wedge F_2).$$

4. Применить закон исключения третьего:

$$F = (\overline{F_2} \wedge F_1) \vee (\overline{F_1} \wedge F_2).$$

Всякая формула алгебры высказываний обладает как дизъюнктивной, так и конъюнктивной нормальными формами. У формулы алгебры логики существует неограниченно много как дизъюнктивных, так и конъюнктивных нормальных форм. Одни из них более громоздкие и сложные, другие – более простые.

Совершенные нормальные формы

Если каждая элементарная конъюнкция (или элементарная дизъюнкция) формулы содержит символы всех пропозициональных переменных, то такая формула называется *совершенной*.

Существуют *совершенные дизъюнктивные нормальные формы* формулы (СДНФ) и *совершенные конъюнктивные нормальные формы* формулы (СКНФ).

Алгоритм преобразования ДНФ к виду СДНФ

Шаг 1. Если в элементарную конъюнкцию F не входит подформула F_i или $\overline{F_i}$, то дополнить элементарную конъюнкцию высказыванием $F_i \vee \overline{F_i}$ и выполнить преобразование формулы по закону дистрибутивности:

$$F (F_i \vee \overline{F_i}) = F F_i \vee F \overline{F_i}.$$

При этом знак конъюнкции в формулах можно опускать, т. е. запись $F_1 \overline{F_2}$ соответствует записи $F_1 \wedge \overline{F_2}$, или $F_1 \cdot \overline{F_2}$.

Шаг 2. Если в элементарную конъюнкцию F не входит подформула F_j или $\overline{F_j}$, то повторить шаг 1.

Шаг 3. Упрощаем полученную формулу, используя равносильности:

$$F \wedge F = F; \quad F \vee F = F.$$

Пример

Необходимо преобразовать формулу

$$F = F_1 \overline{F_2} \vee F_1 \overline{F_3} F_4 \vee F_1 F_2 F_3 \overline{F_4}$$

к виду СДНФ:

$$F = F_1 \overline{F_2} (F_3 \vee \overline{F_3}) \vee F_1 \overline{F_3} F_4 (F_2 \vee \overline{F_2}) \vee F_1 F_2 F_3 \overline{F_4};$$

$$F = F_1 \overline{F_2} F_3 \vee F_1 \overline{F_2} \overline{F_3} \vee F_1 F_2 \overline{F_3} F_4 \vee F_1 \overline{F_2} \overline{F_3} F_4 \vee F_1 F_2 F_3 \overline{F_4};$$

$$F = F_1 \overline{F_2} F_3 (F_4 \vee \overline{F_4}) \vee F_1 \overline{F_2} \overline{F_3} (F_4 \vee \overline{F_4}) \vee \\ \vee F_1 F_2 \overline{F_3} F_4 \vee F_1 \overline{F_2} \overline{F_3} F_4 \vee F_1 F_2 F_3 \overline{F_4};$$

$$F = F_1 \overline{F_2} F_3 F_4 \vee F_1 \overline{F_2} F_3 \overline{F_4} \vee F_1 \overline{F_2} \overline{F_3} F_4 \vee F_1 \overline{F_2} \overline{F_3} \overline{F_4} \vee \\ \vee F_1 F_2 \overline{F_3} F_4 \vee F_1 \overline{F_2} \overline{F_3} F_4 \vee F_1 F_2 F_3 \overline{F_4};$$

$$F = F_1 \overline{F_2} F_3 F_4 \vee F_1 \overline{F_2} F_3 \overline{F_4} \vee F_1 \overline{F_2} \overline{F_3} F_4 \vee F_1 \overline{F_2} \overline{F_3} \overline{F_4} \vee \\ \vee F_1 F_2 \overline{F_3} F_4 \vee F_1 F_2 F_3 \overline{F_4}.$$

Алгоритм преобразования КНФ к виду СКНФ

Шаг 1. Если в элементарную дизъюнкцию F не входит подформула F_i или $\overline{F_i}$, то дополнить элементарную дизъюнкцию высказыванием $F_i \overline{F_i}$ и выполнить преобразование формулы по закону дистрибутивности:

$$F \vee F_i \overline{F_i} = (F \vee F_i) (F \vee \overline{F_i}).$$

Шаг 2. Если в элементарную конъюнкцию F не входит подформула F_j или $\overline{F_j}$, то повторить шаг 1.

Шаг 3. Упрощаем полученную формулу, используя равносильности:

$$F \wedge F = F; \quad F \vee F = F.$$

Пример

Необходимо преобразовать формулу $F = (F_1 \vee F_2) (\overline{F_1} \vee \overline{F_2} \vee F_3)$ к виду СКНФ:

$$F = (F_1 \vee F_2 \vee F_3 \overline{F_3}) (\overline{F_1} \vee \overline{F_2} \vee F_3);$$

$$F = (F_1 \vee F_2 \vee F_3) (F_1 \vee F_2 \wedge \overline{F_3}) (\overline{F_1} \vee \overline{F_2} \vee F_3).$$

Совершенные нормальные формы формул удобно записывать, используя таблицы истинности.

Элементарные конъюнкции СДНФ формируются для значений формулы 1. Число элементарных конъюнкций равно числу истинных значений формулы. Пропозициональные переменные, входящие в элементарную конъюнкцию, записываются без изменений, если их значение равно 1, и с логической связкой отрицание – если их значение равно 0.

СДНФ для всякой логической функции единственна. Для тождественно ложной функции СДНФ не существует.

Элементарные дизъюнкции СКНФ формируются для значений формулы 0. Число элементарных дизъюнкций равно числу ложных значений формулы. Пропозициональные переменные, входящие в элементарную дизъюнкцию, записываются без изменений, если их значение равно 0, и с логической связкой отрицание – если их значение равно 1.

СКНФ для всякой логической функции единственна. Единственная функция, не имеющая СКНФ, – тождественно истинная.

Пример

Необходимо записать СДНФ и СКНФ для функции, заданной таблицей истинности (табл. 21).

Таблица 21

<i>A</i>	<i>B</i>	<i>C</i>	<i>F (A, B, C)</i>
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Формула СДНФ имеет вид

$$F(A, B, C) = \overline{A}\overline{B}\overline{C} \vee \overline{A}B\overline{C} \vee \overline{A}B\overline{C} \vee ABC.$$

Формула СКНФ имеет вид

$$F(A, B, C) = (A \vee B \vee \overline{C})(A \vee \overline{B} \vee C)(\overline{A} \vee B \vee \overline{C})(\overline{A} \vee \overline{B} \vee C).$$

1.5. Функционально полные системы элементарных булевых функций

Принцип суперпозиций. Замкнутые классы логических функций

Принцип суперпозиции заключается в подстановке булевых функций вместо аргументов в другую булеву функцию. С помощью принципа суперпозиции любая булева функция может быть представлена как некоторая комбинация функций двух переменных.

Суперпозицией булевых функций f_0 и f_1, \dots, f_n называется функция $f(x_1, \dots, x_m) = f_0(g_1(x_1, \dots, x_m), \dots, g_k(x_1, \dots, x_m))$, где каждая из функций $g_i(x_1, \dots, x_m)$ совпадает либо с одной из переменных (тождественная функция), либо – с одной из функций f_1, \dots, f_n .

Пример

Функция $f(x, y) = \neg(x \& y)$ является суперпозицией функций \neg и $\&$.

Функция $g(x, y) = x \oplus (x \vee y)$ является суперпозицией функций \oplus и \vee .

Функция $h(x, y, z) = (x \& y) \oplus z$ – суперпозиция функций \oplus и $\&$.

Рассмотрим множество A логических функций, обладающих некоторым свойством. Пусть $G(Y_1, Y_2, \dots, Y_k) \in A$ и $F_i(X_1, X_2, \dots, X_n) \in A$, $i = 1, 2, \dots, k$. Произведем суперпозицию функций G и F :

$$G[F_1(X_1, \dots, X_n), \dots, F_k(X_1, \dots, X_n)] = H(X_1, \dots, X_n).$$

Если $H(X_1, \dots, X_n) \in A$, то A называется *замкнутым классом логических функций* по отношению к рассматриваемому свойству.

Таким образом, множество A логических функций называют *замкнутым классом*, если любая суперпозиция функций из A снова принадлежит A .

Полином Жегалкина

Представление функции в системе $\{\wedge, \oplus, 1\}$ называется полиномом Жегалкина данной функции. Полином Жегалкина, и притом единственный, существует для всякой логической функции.

Свойства функций системы Жегалкина

1. Ассоциативность:

а) $x(yz) = (xy)z$;

б) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$.

2. Коммутативность:

а) $xy = yx$;

б) $x \oplus y = y \oplus x$.

3. Дистрибутивность: $x(y \oplus z) = xy \oplus xz$.

4. Попарное сокращение: $x \oplus x = 0$.

5. Свойство констант: $x \oplus 0 = x$.

Функция, у которой полином Жегалкина имеет вид $\sum \alpha_i x_i \oplus \gamma$, где α_i, γ равны 0 или 1, называется *линейной*.

Связь функций системы Жегалкина с булевыми функциями

1. Выражение отрицания через функции системы Жегалкина:

$$\bar{x} = x \oplus 1.$$

2. Выражение дизъюнкции через функции системы Жегалкина:

$$x \vee y = xy \oplus x \oplus y.$$

Свойства, по которым функции образуют замкнутые классы

Различают пять «замечательных» свойств, по которым логические функции образуют замкнутые классы.

1. *Свойство сохранять 0*. Булева функция сохраняет 0, если на нулевом наборе она принимает нулевое значение. Нулевой набор состоит из n нулей, где n – число аргументов булевой функции.

Например, функции $X_1 \wedge X_2$, $X_1 + X_2$, $X_1 \oplus X_2$ сохраняют 0, а функции $X_1 \rightarrow X_2$, $X_1 \sim X_2$, $X_1 | X_2$ не сохраняют.

Через C_0 обозначим класс всех функций, сохраняющих 0, т. е. $C_0 = \{F(X_1, \dots, X_n) \mid F(0, \dots, 0) = 0\}$.

Класс C_0 является замкнутым.

2. *Свойство сохранять 1*. Булева функция сохраняет 1, если на единичном наборе она принимает единичное значение. Набор называется единичным, если он состоит только из единиц.

Например, функции $X_1 \wedge X_2$, $X_1 + X_2$, $X_1 \rightarrow X_2$ сохраняют единицу, а $X_1 \oplus X_2$, $X_1 | X_2$ не сохраняют.

Через C_1 обозначим класс всех функций, сохраняющих 1, т. е. $C_1 = \{F(X_1, \dots, X_n) \mid F(1, \dots, 1) = 1\}$.

Класс C_1 является замкнутым.

3. *Самодвойственность.* Функция называется самодвойственной, если имеет место равенство $F(X_1, X_2, \dots, X_n) = \overline{F(\overline{X_1}, \overline{X_2}, \dots, \overline{X_n})}$, где $\overline{F(\overline{X_1}, \overline{X_2}, \dots, \overline{X_n})}$ – двойственная функция по отношению к функции F . Двойственная функция обозначается F^* .

Пример

$F = X_1X_2 + X_1X_3 + X_2X_3$ является самодвойственной функцией.

Действительно,

$$\begin{aligned} F^* &= \overline{\overline{X_1} \overline{X_2} + \overline{X_1} \overline{X_3} + \overline{X_2} \overline{X_3}} = \overline{\overline{X_1} \overline{X_2}} \cdot \overline{\overline{X_1} \overline{X_3}} \cdot \overline{\overline{X_2} \overline{X_3}} = \\ &= (X_1 + X_2)(X_1 + X_3)(X_2 + X_3) = X_1X_2 + X_1X_3 + X_2X_3 = F. \end{aligned}$$

Класс всех самодвойственных функций обозначим C_S .

Согласно определению самодвойственная функция на противоположных наборах аргументов принимает противоположные значения.

Рассмотрим, что происходит с таблицей двойственной функции. Замена набора (x_1, \dots, x_n) на $(\neg x_1, \dots, \neg x_n)$ соответствует «переворачиванию» таблицы. Действительно, наборы (x_1, \dots, x_n) и $(\neg x_1, \dots, \neg x_n)$ расположены симметрично относительно середины таблицы. Теперь остается применить операцию «отрицания» к результату функции, т. е. поменять 0 на 1 и 1 на 0. Таким образом вектор значений функции, двойственной к исходной, получается из вектора исходной функции переворачиванием и заменой 0 на 1, а 1 на 0.

X	Y	$F = X \wedge Y$	$\neg X \wedge \neg Y$	$\neg(\neg X \wedge \neg Y)$
0	0	0	1	0
0	1	0	0	1
1	0	0	0	1
1	1	1	0	1

Пример

Функции $x \& y$ и $x \vee y$, задаваемые векторами значений $(0, 0, 0, 1)$ и $(0, 1, 1, 1)$, двойственны друг к другу. Также двойственными являются $x \oplus y$ и $x \equiv y$, задаваемые векторами $(0, 1, 1, 0)$ и $(1, 0, 0, 1)$. Каж-

дая из функций x и $\neg x$ (векторы $(0, 1)$ и $(1, 0)$ соответственно) двойственна сама себе (табл. 22).

Таблица 22

$F_1 = X \wedge Y$	F_1^*	$F_2 = X \vee Y$	F_2^*	$F_3 = X \leftrightarrow Y$	F_3^*	$F_4 = X \oplus Y$	F_4^*
0	0	0	0	0	1	1	0
0	1	1	0	1	0	0	1
0	1	1	0	1	0	0	1
1	1	1	1	0	1	1	0

C_S – замкнутый класс.

4. *Монотонность.* Логическая функция называется *монотонной*, если при любом возрастании набора значение этой функции не убывает, т. е. функция $f(x_1, \dots, x_n)$ монотонна, если для любых двоичных наборов δ и τ длины n , при условии $\delta \leq \tau$, выполняется условие $f(\delta) \leq f(\tau)$. При этом рассматриваются только сравнимые наборы. Примеры монотонных функций: XY , $X + Y$. Примеры функций, не обладающих свойством монотонности: $X \sim Y$, $X \rightarrow Y$, $X \oplus Y$. Класс всех монотонных функций обозначим C_M .

Критерий сравнения двух наборов аргументов состоит в следующем. Если значение каждого аргумента одного набора больше или равно значению того же аргумента второго набора, то говорят, что первый набор не меньше второго. При этом предполагается, что $0 \geq 0$; $1 \geq 0$; $1 \geq 1$.

Пример

Наборы: $(1, 1, 0, 1) \geq (0, 1, 0, 1)$.

Не всякие наборы являются сравнимыми. Наборы $(0, 1)$ и $(1, 0)$ или $(0, 1, 0, 1)$ и $(1, 0, 1, 0)$ несравнимы.

Линейность. Логическая функция называется *линейной*, если она может быть представлена полиномом первой степени (т. е. без конъюнкций).

Например, $X \oplus Y$, $X \sim Y = 1 \oplus X \oplus Y$ – линейные функции. Класс всех линейных функций обозначим C_L .

По свойству линейности функции образуют замкнутый класс.

Пример

Необходимо исследовать заданную функцию $f(x, y, z) = x \vee y \vee z$ на принадлежность к классам C_0 , C_1 , C_S , C_M , C_L .

Решение:

1. Проверим на принадлежность к классу C_0 :

$$f(0, 0, 0) = 0 \vee 0 \vee 0 = 0, \text{ следовательно, } f(x, y, z) \in C_0.$$

2. Проверим на принадлежность к классу C_1 :

$$f(1, 1, 1) = 1 \vee 1 \vee 1 = 1, \text{ следовательно, } f(x, y, z) \in C_1.$$

3. Проверим на принадлежность к классу C_S . Получим двойственную функцию

$$f^*(x, y, z) = \overline{f(\bar{x}, \bar{y}, \bar{z})} = \overline{\bar{x} \vee \bar{y} \vee \bar{z}} = xyz.$$

$$f^*(x, y, z) \neq f(x, y, z), \text{ следовательно } f(x, y, z) \notin C_S.$$

4. Проверим на принадлежность к классу C_M . Составим таблицу истинности для заданной функции $f(x, y, z) = x \vee y \vee z$ (табл. 23).

Значение функции при возрастании набора значений на сравнимых наборах не убывает, следовательно, $f(x, y, z) \in C_M$.

5. Проверим на принадлежность к классу C_L . Для исследования функции на линейность необходимо найти многочлен Жегалкина, его поиск произведем методом неопределенных коэффициентов.

Выпишем общий вид многочлена Жегалкина для трех переменных:

$$f(x, y, z) = A_0 \oplus A_1x \oplus A_2y \oplus A_3z \oplus A_4xy \oplus A_5yz \oplus A_6xz \oplus A_7xyz.$$

Таблица 23

x	y	z	$x \vee y \vee z$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Затем подставим искомые значения переменных в многочлен и приравняем его к значению функции на данном наборе. Получаем следующую систему уравнений:

$$\begin{aligned}
 f(0, 0, 0) &= A_0 = 0; \\
 f(0, 0, 0) &= A_0 \oplus A_3 = 1; \\
 f(0, 0, 0) &= A_0 \oplus A_2 = 1; \\
 f(0, 0, 0) &= A_0 \oplus A_2 \oplus A_3 \oplus A_5 = 1; \\
 f(0, 0, 0) &= A_0 \oplus A_1 = 1; \\
 f(0, 0, 0) &= A_0 \oplus A_1 \oplus A_3 \oplus A_6 = 1; \\
 f(0, 0, 0) &= A_0 \oplus A_1 \oplus A_2 \oplus A_4 = 1; \\
 f(0, 0, 0) &= A_0 \oplus A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7 = 1.
 \end{aligned}$$

Получаем следующие значения коэффициентов:

$$A_0 = 0, A_1 = A_2 = A_3 = A_4 = A_5 = A_6 = A_7 = 1.$$

Полином Жегалкина $-f(x, y, z) = x \oplus y \oplus z \oplus xy \oplus yz \oplus xz \oplus xyz$.

$f(x, y, z) \notin C_L$, так как полином не является полиномом первой степени (содержит конъюнкции переменных).

Признаки функциональной полноты

Система элементарных булевых функций называется *функционально полной* (или же *полной системой функций*), если произвольную булеву функцию можно представить суперпозицией функций этой системы. Полная система функций образует базис. Минимальным базисом называется такой, в котором при удалении хотя бы одной функции, образующей этот базис, нарушается его полнота.

Сформулируем *I признак функциональной полноты*. Если все функции функционально полной системы Σ можно выразить через функции системы Σ^* , то Σ^* является функционально полной системой.

Примеры функционально полных систем

- $\Sigma_1 = \{\wedge, \neg\}$.

Полнота следует из возможности выразить дизъюнкцию через функции этой системы. Действительно, $x_1 \vee x_2 = \neg(\overline{x_1} \wedge \overline{x_2})$.

- $\Sigma_2 = \{\vee, \neg\}$.

Полнота следует из возможности выразить конъюнкцию через функции этой системы: $x_1 \wedge x_2 = \neg(\overline{x_1} \vee \overline{x_2})$.

- $\Sigma_3 = \{\}\}$.

Штрих Шеффера представляет функционально полную систему, так как

$$\overline{x} = x | x, x_1 \wedge x_2 = \overline{x_1 | x_2} = (x_1 | x_2) | (x_1 | x_2),$$

т. е. Σ_3 сводится к Σ_1 , а Σ_1 функционально полна.

$$4. \Sigma_4 = \{\downarrow\}.$$

Система функционально полна, так как

$$\bar{x} = x \downarrow x, x_1 \vee x_2 = \overline{x_1 \downarrow x_2} = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2),$$

т. е. Σ_4 сводится к Σ_2 , а Σ_2 функционально полна.

$$5. \Sigma_5 = \{\wedge, \oplus, 1\}.$$

Система функционально полна, так как $\bar{x} = x \oplus 1$, и Σ_1 функционально полна.

Сформулируем *II признак функциональной полноты*.

Для того чтобы система булевых функций была функционально полной, необходимо и достаточно, чтобы эта система включала:

- хотя бы одну функцию, не сохраняющую 0;
- хотя бы одну функцию, не сохраняющую 1;
- хотя бы одну несамодвойственную функцию;
- хотя бы одну немонотонную функцию;
- хотя бы одну нелинейную функцию.

Доказательство этой теоремы основано на том, что суперпозиция любого числа функций, образующих замкнутый класс, представляет собой функцию этого же класса. Можно предположить, что наибольшее число функций, образующих базис, равно пяти. Однако ввиду того, что многие булевы функции удовлетворяют одновременно нескольким требованиям, предъявляемым теоремой о функциональной полноте, количество независимых булевых функций, образующих минимальный базис, меньше пяти.

Рассмотрим некоторые функционально полные системы.

1. $F = X|Y$. Функция (штрих Шеффера) не обладает ни одним из «замечательных» свойств, следовательно, она одна образует минимальный базис.

2. $F = X \downarrow Y$. Логическая функция – «стрелка Пирса»: так же, как и «штрих Шеффера» эта функция не обладает ни одним из указанных свойств и поэтому образует минимальный базис.

3. $F_1 = X \rightarrow Y$ и $F_2 = 0$ или $F_1 = Y \rightarrow X$ и $F_2 = 0$.

4. $F_1 = X \oplus Y$, $F_2 = X \wedge Y$, $F_3 = 1$. Функции этого базиса входят в полином Жегалкина.

Из всего многообразия возможных функционально полных систем булевых функций в технике наибольшее распространение получил базис, содержащий три функции: конъюнкцию, дизъюнкцию и отри-

вание. Этот базис не является минимальным, но использование всех трех указанных функций совместно с константами 0 и 1 позволяет сравнительно легко строить сложные логические устройства на электронных элементах.

Пример

Исследовать систему функций $\{x \oplus y; x \vee y; 1\}$ на функциональную полноту.

Решение

Составим таблицу Поста следующего вида, представленную в табл. 24. Знак «+», стоящий на пересечении строк и столбцов, показывает, что функция принадлежит соответствующему классу.

Таблица 24

Функция	C_0	C_1	C_S	C_M	C_L
$f_1 = x \oplus y$					
$f_2 = x \vee y$					
$f_3 = 1$					

Построим таблицу истинности для функций, входящих в заданную систему и двойственных функций к ним.

Таблица истинности для системы функций $\{x \oplus y; x \vee y; 1\}$ – табл. 25.

Таблица 25

x	y	$f_1 = x \oplus y$	$f_2 = x \vee y$	$f_3 = 1$	f_1^*	f_2^*	f_3^*
0	0	0	0	1	1	0	0
0	1	1	1	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	1	1	1	1	0

$$f(0, 0) = 0 \oplus 0 = 0 \Rightarrow f_1 \in C_0.$$

$$f(1, 1) = 1 \oplus 1 = 0 \Rightarrow f_1 \notin C_1.$$

$$f_1^* \neq f_1 \Rightarrow f_1 \notin C_S.$$

Условие монотонности нарушается на наборах (1, 0) и (1, 1) $\Rightarrow f_1 \notin C_M$.

$f_1(x, y)$ – линейная функция, так как многочлен Жегалкина $x \oplus y$ линеен $\Rightarrow f_1 \in C_L$.

$$f_2(0, 0) = 0 \vee 0 = 0 \Rightarrow f_2 \in C_0.$$

$$f_2(1, 1) = 1 \vee 1 = 1 \Rightarrow f_2 \in C_1.$$

$$f_2^* \neq f_2 \Rightarrow f_2 \notin C_S.$$

Условие монотонности не нарушается $\Rightarrow f_2 \in C_M$.

$f_2(x, y) = x \vee y$ – нелинейная функция, так как соответствующий ей многочлен Жегалкина $x \oplus y \oplus xy$ нелинеен $\Rightarrow f_2 \notin C_L$.

$$f_3(0, 0) = 1 \Rightarrow f_3 \notin C_0.$$

$$f_3(1, 1) = 1 \Rightarrow f_3 \in C_1.$$

$$f_3^* \neq f_3 \Rightarrow f_3 \notin C_S.$$

Условие монотонности не нарушается $\Rightarrow f_3 \in C_M$.

$f_3(x, y) = 1$ – константа единица – линейная функция, так как соответствующий ей многочлен Жегалкина 1 линеен $\Rightarrow f_3 \in C_L$.

Проанализировав таблицу Поста, построим табл. 26.

Таблица 26

Функция	C_0	C_1	C_S	C_M	C_L
$f_1 = x \oplus y$	+	–	–	–	+
$f_2 = x \vee y$	+	+	–	+	–
$f_3 = 1$	–	+	–	+	+

Можно сделать вывод, что все условия теоремы Поста выполнены и заданная система $\{x \oplus y; x \vee y; 1\}$ является функционально полной.

Практические задания

1. Записать высказывания с использованием пропозициональных переменных и логических операций: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$:

(а) *Студент Иванов не успевает по математической логике тогда и только тогда, когда он не изучает английский язык.*

(б) *Если мистер Смит счастлив, то миссис Смит несчастлива, и если мистер Смит несчастлив, то миссис Смит счастлива.*

(в) *Алиса слушает музыку только в том случае, если эта музыка – джаз.*

(г) *Если студент ложится спать поздно и пьет на ночь кофе, то утром он встает в дурном расположении духа или с головной болью.*

(д) Для того чтобы целое число x было нечетным, достаточно, чтобы x было простым.

2. Записать высказывание в виде формулы логики высказываний. Построить отрицание данного высказывания в виде формулы, не содержащей внешних знаков отрицания. Перевести на естественный язык.

(а) Он и жнец, и швец, и на дуде игрец.

(б) Если стальное колесо нагреть, то его диаметр увеличится.

(в) Если функция нечетная, то ее график симметричен относительно начала координат.

(г) Если ряд сходится, то его общий член стремится к нулю.

(д) Если составить алгоритм и написать программу, то можно решить эту задачу.

3. Записать на языке логики высказываний выражения, соответствующие следующему условию:

Три студента Александр, Владимир и Дмитрий перед сдачей зачета по физкультуре тренировались в прыжках в высоту. После того как была установлена планка, Александр предложил пари Владимиру, сказав: «Мой прыжок будет удачным в том и только в том случае, если твой окажется неудачным». Аналогичное пари Владимир предложил Дмитрию, а Дмитрий Александру.

Необходимо, используя таблицу истинности, определить, возможна ли ситуация, чтобы никто из них не проиграл пари? Чтобы хоть кто-нибудь выиграл пари?

4. Определить значения истинности высказываний A, B, C, D, E, F , если высказывания исходят из условий:

(а) $A \wedge (2 \cdot 2 = 4) = 1$;

(б) $B \vee (2 \cdot 2 = 5) = 1$;

(в) $C \vee (2 \cdot 2 = 4) = 1$;

(г) $D \wedge (2 \cdot 2 = 5) = 0$;

(д) $\overline{E} \wedge (2 \cdot 2 = 4) = 0$;

(е) $\overline{F} \wedge (2 \cdot 2 = 5) = 0$.

5. Определить значения истинности высказываний A, B, C, D, E, F , если высказывания (а)–(в) ложны, а высказывания (г)–(е) истинны:

(а) если 4 – четное число, то A ;

(б) если B , то 6 – четное число;

(в) если $2 \cdot 2 = 4$, то \overline{C} ;

(г) если $3 \cdot 2 = 6$, то \overline{D} ;

(д) если \bar{D} , то $2 \cdot 2 = 6$;

(е) если 3 – четное число, то F .

6. Определить значения истинности высказываний A, B, C, D, E, F , если высказывания (а)–(в) истинны, а высказывания (г)–(е) ложны:

(а) $A \leftrightarrow (2 < 3)$;

(б) $B \leftrightarrow (2 > 3)$;

(в) $(6 \leq 7) \leftrightarrow \bar{C}$;

(г) $D \leftrightarrow (2 < 3)$;

(д) $E \leftrightarrow (2 > 3)$;

(е) $(6 \leq 7) \leftrightarrow \bar{F}$.

7. Из трех данных высказываний A, B, C необходимо построить такое высказывание, которое:

(а) истинно тогда и только тогда, когда все данные высказывания истинны;

(б) ложно тогда и только тогда, когда все данные высказывания ложны;

(в) ложно тогда и только тогда, когда все данные высказывания истинны;

(г) истинно тогда и только тогда, когда истинны высказывания A и B ;

(д) ложно тогда и только тогда, когда ложны высказывания A и B ;

(е) ложно тогда и только тогда, когда ложно лишь высказывание C ;

(ж) ложно тогда и только тогда, когда истинны высказывания A и B ;

(з) ложно тогда и только тогда, когда все данные высказывания либо истинны, либо ложны.

8. Определить логическое значение последнего высказывания, исходя из логических значений всех предыдущих высказываний:

(а) $B \leftrightarrow A = 0, B \rightarrow A = 1, A \rightarrow B = \dots$

(б) $A \leftrightarrow B = 0, (A \rightarrow B) \wedge (\bar{A} \rightarrow \bar{B}) = \dots$

(в) $A \wedge B = 0, A \leftrightarrow B = 0, A \rightarrow B = 1, A = \dots$

(г) $A \leftrightarrow B = 0, \bar{B} \rightarrow A = \dots$

(д) $A \leftrightarrow B = 0, A \rightarrow B = 1, ((\bar{A} \rightarrow B) \leftrightarrow A) = \dots$

9. Для каждого из помещенных ниже высказываний определить, достаточно ли приведенных сведений, чтобы установить его логиче-

ское значение (если достаточно, то указать это значение; если недостаточно, то продемонстрировать на примерах, что возможны и одно и другое истинностные значения):

(а) $(A \rightarrow B) \rightarrow (\bar{A} \rightarrow \bar{B}), A = 0;$

(б) $\overline{(B \rightarrow A)} \leftrightarrow \overline{(A \vee C)}, A = 1;$

(в) $\overline{(A \vee B)} \leftrightarrow (\bar{B} \wedge \bar{A}), B = 0;$

(г) $(A \rightarrow B) \rightarrow (\bar{A} \rightarrow \bar{B}), B = 1;$

(д) $(A \leftrightarrow B) \vee (A \wedge C), A = 0.$

10. Существуют ли три таких высказывания A, B, C , чтобы для них одновременно выполнялись следующие условия:

(а) $(A \wedge B) = 1, A \wedge C = 0, A \wedge B \wedge \bar{C} = 0;$

(б) $(B \rightarrow A) = 0, A \vee C = 0, A \leftrightarrow (B \wedge \bar{C}) = 0;$

(в) $B \vee C = 0, \bar{C} \rightarrow A = 0, A \rightarrow B = 0;$

(г) $B \vee C = 1, \bar{C} \rightarrow A = 1, A \rightarrow B = 0;$

(д) $(A \wedge B) = 0, A \wedge C = 1, A \wedge B \wedge \bar{C} = 1?$

11. Определить, какие переменные являются существенными, а какие фиктивными в следующих формулах:

(а) $(X \wedge Y) \vee \bar{X};$

(б) $(Y \wedge X) \vee (\bar{Y} \wedge Z);$

(в) $(X \rightarrow (Y \rightarrow Z)) \rightarrow ((X \rightarrow Y) \rightarrow (X \rightarrow Z));$

(г) $X \wedge Y \wedge \bar{Z} \vee \overline{X \wedge \bar{Y} \wedge Z} \vee Y \wedge \bar{Z} \vee \bar{X} \wedge Y \wedge \bar{Z};$

(д) $((X \vee Y) \rightarrow X \wedge Y) \oplus (X \rightarrow Y) \rightarrow (Y \rightarrow X);$

(е) $(X \wedge Y \oplus (X \rightarrow Y)) \rightarrow (X \leftrightarrow X \wedge Y);$

(ж) $(X \rightarrow \bar{Y}) \oplus (Y \rightarrow \bar{Z}) \oplus (Y \rightarrow Z).$

12. Необходимо выяснить, какие переменные следующих функций являются существенными, а какие несущественными, и выразить $f(x, y, z)$ формулой, содержащей только существенные переменные:

(а) $f(x, y, z) = 01011010;$

(б) $f(x, y, z) = 10101010;$

(в) $f(x, y, z) = 11110000;$

(г) $f(x, y, z) = 01111111;$

(д) $f(x, y, z) = 01010101.$

13. Упростить следующие формулы при помощи эквивалентных преобразований:

(а) $\bar{\bar{X}} \wedge \bar{\bar{Y}} \vee (X \rightarrow Y) \wedge X;$

- (б) $X \wedge Z \vee X \wedge \bar{Z} \vee Y \wedge Z \vee \bar{X} \wedge Y \wedge Z$;
 (в) $(\bar{\bar{X}} \vee \bar{Y} \rightarrow X \vee Y) \wedge Y$;
 (г) $\overline{(X \rightarrow Y)} \wedge \overline{(Y \rightarrow X)}$;
 (д) $(X \rightarrow \bar{Y}) \vee (\bar{X} \vee \bar{Y})$.

14. Составить таблицы истинности для следующих формул и указать классы логических формул: выполнимые, опровержимые, тождественно-истинные (тавтологии), тождественно-ложные (противоречия):

- (а) $P \wedge (Q \wedge (\bar{P} \vee \bar{Q}))$;
 (б) $(P \rightarrow Q) \rightarrow ((P \rightarrow \bar{Q}) \rightarrow \bar{P})$;
 (в) $(P \wedge (Q \vee \bar{P})) \wedge ((\bar{Q} \rightarrow \bar{P}) \vee Q)$;
 (г) $((P \rightarrow Q) \rightarrow Q) \rightarrow Q$;
 (д) $((P \vee \bar{Q}) \wedge (Q \vee R)) \vee \bar{R} \vee Q$;
 (е) $(P \wedge (Q \vee R)) \rightarrow ((R \rightarrow (P \rightarrow Q)) \leftrightarrow (Q \rightarrow (R \rightarrow P)))$.

15. Доказать, что следующие формулы выполнимы, указав значения входящих в них пропозициональных переменных, при которых эти формулы превращаются в истинные высказывания:

- (а) $(Q \rightarrow (P \wedge R)) \wedge \overline{((P \vee R) \rightarrow Q)}$;
 (б) $\overline{((P \leftrightarrow \bar{Q}) \vee R)} \wedge Q$;
 (в) $(P \rightarrow Q) \rightarrow (R \rightarrow Q) \rightarrow (R \rightarrow P) \rightarrow (P \rightarrow Q)$;
 (г) $((Q \rightarrow \bar{P}) \rightarrow P) \rightarrow (P \rightarrow (\bar{P} \rightarrow Q))$;
 (д) $(P \rightarrow ((Q \rightarrow R) \rightarrow R)) \rightarrow (P \rightarrow (Q \rightarrow R)) \rightarrow (P \rightarrow R)$;
 (е) $((P \rightarrow Q) \wedge (Q \rightarrow R)) \rightarrow (\bar{R} \rightarrow \bar{P})$.

16. Доказать, что следующие формулы опровержимы, указав значения входящих в них пропозициональных переменных, при которых эти формулы превращаются в ложные высказывания:

- (а) $\overline{(((\bar{P} \rightarrow P) \rightarrow ((\bar{Q} \rightarrow \bar{P}) \rightarrow ((\bar{Q} \rightarrow P) \rightarrow Q)))}$;
 (б) $\overline{((P \leftrightarrow \bar{Q}) \vee R)} \rightarrow Q$;
 (в) $((P \rightarrow Q) \rightarrow (R \rightarrow Q)) \rightarrow (R \rightarrow P) \rightarrow (P \rightarrow Q)$;
 (г) $((Q \rightarrow \bar{P}) \rightarrow P) \rightarrow (P \rightarrow (\bar{P} \rightarrow Q))$;
 (д) $(P \rightarrow ((Q \rightarrow \bar{R}) \rightarrow \bar{R})) \rightarrow (P \rightarrow (Q \rightarrow \bar{R})) \rightarrow (P \rightarrow \bar{R})$;
 (е) $\overline{((P \wedge \bar{Q}) \wedge (Q \rightarrow R) \wedge (R \vee S)) \rightarrow (\bar{S} \wedge Q)}$.

17. Установить при помощи равносильных преобразований, являются ли данные формулы тождественно-истинными (тавтологиями):

(а) $(Q \rightarrow R) \rightarrow ((P \vee Q) \rightarrow (P \vee R))$;

(б) $P \rightarrow (Q \rightarrow ((P \vee Q) \rightarrow (P \wedge Q)))$;

(в) $(Q \rightarrow R) \rightarrow (P \vee Q) \rightarrow (P \vee R)$;

(г) $X \rightarrow (Y \rightarrow Z) \rightarrow (XY \rightarrow Z)$;

(д) $XY(XY \rightarrow Z)\bar{Z}$;

(е) $((P \rightarrow Q) \vee R) \leftrightarrow (P \rightarrow (Q \vee R))$?

18. При помощи равносильных преобразований доказать равносильность следующих формул:

(а) $((P \rightarrow (Q \wedge R)) \equiv (P \rightarrow Q) \vee (P \rightarrow R))$;

(б) $((P \rightarrow (Q \leftrightarrow R)) \equiv (P \rightarrow Q) \leftrightarrow (P \rightarrow R))$;

(в) $((P \wedge (Q \leftrightarrow R)) \equiv (P \wedge Q) \leftrightarrow (P \wedge R))$.

19. Упростить данную систему истинных высказываний, т. е. найти логически эквивалентную ей систему, состоящую из меньшего числа не более сложных высказываний:

(а) $Z \rightarrow (X \vee Y), (Y \wedge Z) \rightarrow X, (Y \wedge X) \rightarrow Z$;

(б) $X \rightarrow (Y \vee Z), Y \rightarrow (X \vee Z), (X \wedge Y) \rightarrow Z$;

(в) $X \rightarrow Y, X \rightarrow (Y \vee Z), Y \rightarrow Z$;

(г) $P \rightarrow Q, Q \rightarrow \bar{P}, R \rightarrow P$;

(д) $X \vee Y, Y \rightarrow Z, (Y \wedge Z) \rightarrow X$;

(е) $P \rightarrow (Q \vee R), W \rightarrow (S \vee T), R \rightarrow (Q \vee \bar{P}), (W \wedge T) \rightarrow \bar{S}$.

20. Привести формулы к дизъюнктивной и конъюнктивной нормальным формам:

(а) $(X \rightarrow Y) \rightarrow Z$;

(б) $X \leftrightarrow Y \leftrightarrow Z$;

(в) $\overline{X \wedge Y} \vee (X \rightarrow Y)$;

(г) $(X \leftrightarrow Y) \wedge (Y \leftrightarrow Z)$;

(д) $(X \vee Y \vee Z) \wedge (X \rightarrow Y)$;

(е) $(X \vee Y) \wedge (Y \vee Z) \rightarrow (X \vee Z)$.

21. Найти СДНФ, эквивалентную данным формулам, при помощи эквивалентных преобразований, результат проверить по таблице истинности:

(а) $(X \rightarrow Y) \wedge (Y \rightarrow Z) \rightarrow (X \rightarrow Z)$;

(б) $(X \rightarrow Y) \wedge (Y \rightarrow Z) \wedge (Z \rightarrow X)$;

(в) $X \vee Y \rightarrow (\bar{X} \rightarrow Y)$;

(г) $\overline{\overline{\overline{(X \wedge Y \rightarrow X)}(X \wedge Y \rightarrow Y)}}$;

(д) $X \wedge (Y \vee \bar{Z}) \wedge (X \vee Y \vee Z)$.

22. Найти СКНФ, эквивалентную данным формулам, при помощи эквивалентных преобразований, результат проверить по таблице истинности:

(а) $\overline{X \rightarrow Y \wedge Z}$;

(б) $X \wedge \bar{Y} \wedge (X \rightarrow Y) \vee Z$;

(в) $X \wedge Y \wedge Z \leftrightarrow X \wedge Y$;

(г) $X \vee Y \rightarrow (X \rightarrow Z)$;

(д) $((X \rightarrow Y) \rightarrow X \vee \bar{Y}) \vee Z$.

23. Доказать, что справедливы следующие логические следования, руководствуясь определением этого понятия:

(а) $P \vee \bar{R} \rightarrow Q \models (P \rightarrow Q) \wedge R$;

(б) $(P \rightarrow Q) \rightarrow R \models P \rightarrow (Q \rightarrow R)$;

(в) $P \rightarrow Q, P \rightarrow \bar{Q} \models \bar{P}$;

(г) $P \rightarrow Q, P \vee R \models (P \vee R) \rightarrow (P \wedge Q)$.

24. Установить, является ли данное рассуждение правильным (проверить, является ли заключение логическим следствием из набора посылок):

(а) *Если человек принял какое-то решение и он правильно воспитан, то он преодолеет все конкурирующие желания. Человек принял решение, но не преодолел конкурирующих желаний. Следовательно, он неправильно воспитан.*

(б) *Если данное явление психическое, то оно обусловлено внешним воздействием на организм. Если оно физиологическое, то оно тоже обусловлено внешним воздействием на организм. Данное явление не психическое и не физиологическое. Следовательно, оно не обусловлено внешним воздействием на организм.*

(в) *Если капиталовложения останутся постоянными, то возрастут правительственные расходы или возникнет безработица. Если правительственные расходы не возрастут, то налоги будут снижены. Если налоги будут снижены и капиталовложения останутся постоянными, то безработица не возрастет. Безработица не возрастет. Следовательно, правительственные расходы возрастут.*

(г) Если философ – дуалист, то он не материалист. Если он не материалист, то он диалектик или метафизик. Он не метафизик. Следовательно, он диалектик или дуалист.

(д) Если Иванов работает, то он получает зарплату. Если же Иванов учится, то он получает стипендию. Но Иванов не получает зарплату или не получает стипендию. Следовательно, он не работает или не учится.

(е) Если $x \neq 0$ и $y \neq 0$, то $x^2 + y^2 > 0$. Если $x = 0$ и $y = 0$, то выражение $(x - y):(x + y)$ не имеет смысла. Неверно, что $x^2 + y^2 > 0$. Следовательно, не имеет смысла выражение $(x - y):(x + y)$.

(ж) Если человек занимается спортом, то он здоров. Если человек здоров, то он счастлив, Этот человек занимается спортом. Значит, он счастлив.

25. Расположить формулы так, чтобы из каждой логически следовали все стоящие после нее:

(а) $P \vee Q, \overline{P \rightarrow (Q \rightarrow P)}, \overline{(\overline{P} \wedge \overline{Q})}, \overline{P} \leftrightarrow Q, \overline{P} \wedge Q;$

(б) $P \rightarrow Q, P \rightarrow (Q \rightarrow (P \wedge Q)), \overline{P} \wedge \overline{Q}, Q \vee \overline{P}, Q \leftrightarrow P.$

26. Методом от противного выяснить, верны ли следующие логические следования:

(а) $F \rightarrow G, K \rightarrow L, F \vee K \models G \vee L;$

(б) $F \rightarrow G, ((F \vee L) \wedge H) \rightarrow M, L \rightarrow H \models ((F \vee L) \wedge G) \rightarrow \overline{M};$

(в) $F \rightarrow G, \overline{K} \rightarrow \overline{L}, S \rightarrow H, \overline{F} \rightarrow \overline{K}, H \rightarrow L \models S \rightarrow G.$

27. Проверить равносильность следующих формул, пользуясь свойством единственности совершенных нормальных форм (приведением их к СДНФ, СКНФ):

(а) $\overline{(x \wedge \overline{y})} \rightarrow (\overline{y} \rightarrow x)$ и $\overline{(x \rightarrow y) \vee x \vee y};$

(б) $\overline{(x \vee (x \rightarrow y)) \wedge y} \rightarrow y$ и $y \wedge \overline{((y \rightarrow x) \vee x)};$

(в) $\overline{(x \rightarrow y) \wedge (y \rightarrow \overline{x})}$ и $y \rightarrow \overline{x};$

(г) $(x \wedge z) \rightarrow y$ и $z \rightarrow y;$

(д) $(x \rightarrow \overline{y}) \rightarrow (y \wedge z)$ и $(x \wedge y) \vee (y \wedge z);$

(е) $(x \rightarrow y) \rightarrow (y \wedge z)$ и $(z \rightarrow \overline{y}) \vee (x \wedge \overline{y}).$

28. Для следующих булевых функций найти представляющий их полином Жегалкина:

(а) $\overline{x}(y\overline{z} \vee \overline{y}z);$

(б) $(x \rightarrow (y \rightarrow \overline{z})) (y\overline{z} \rightarrow x);$

(в) $(x \oplus 1) (y \oplus 1) \overline{z} \vee yz;$

- (г) $\bar{x}\bar{z} \vee (\bar{x}y \vee (\bar{x}y \vee x\bar{y}))z$;
 (д) $(x \vee y \vee z)(x \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee z)$;
 (е) $x(y \rightarrow z) \vee (\bar{x}y \vee x\bar{y})(z \oplus 1)$.

29. Выяснить, верно ли следующее равенство, отыскав полиномы Жегалкина, представляющие булевы функции в обеих частях следующих равенств:

- (а) $x \leftrightarrow \bar{y} = x\bar{y} \vee \bar{x}y$;
 (б) $xyz \vee xy\bar{z} \vee \bar{x}yz = \overline{(\bar{y} \vee \bar{z})xy\bar{z}}$;
 (в) $x \rightarrow (y \rightarrow z) = (x \rightarrow y) \rightarrow (x \rightarrow z)$;
 (г) $xy \rightarrow z = (x \rightarrow z)(y \rightarrow z)$;
 (д) $x\bar{y}z \vee \bar{x}\bar{y}z \vee xy = \bar{y}z \vee xy \oplus 1 \oplus x \oplus \bar{x}$.

30. Проверить, что каждая булева функция от одного аргумента линейна.

31. Найти все линейные булевы функции от двух аргументов. Сколько имеется таких функций?

32. Доказать, что всего имеется 2^{n+1} линейных булевых функций от n аргументов.

33. Доказать, что суперпозиция любого числа линейных булевых функций является снова линейной булевой функцией. Может ли суперпозиция нелинейных функций дать функцию линейную?

34. Для каждой булевой функции от одного аргумента найти двойственную ей булеву функцию.

35. Для каждой булевой функции от двух аргументов найти двойственную ей булеву функцию.

36. Для заданных булевых функций найти двойственные функции и представить найденные функции их в виде СДНФ:

- (а) $\bar{x}\bar{y}\bar{z} \vee x\bar{y}z \vee xyz$;
 (б) $\bar{x}\bar{y}\bar{z} \vee xyz$;
 (в) $\bar{x}yz \vee \bar{x}\bar{y}\bar{z} \vee x\bar{y}z \vee x\bar{y}\bar{z}$;
 (г) $x(y \oplus z) \vee ((\bar{x} \vee y \vee z) \rightarrow \bar{x}yz)$;
 (д) $(z \rightarrow (x \vee y)) \vee x(y \oplus z)$;
 (е) $(z \rightarrow (x \vee y)) \vee x(y \oplus z)$;
 (ж) $xyz \oplus \overline{xy} \oplus yz \oplus z$;
 (з) $\bar{x}y \vee (z \rightarrow x)$;
 (и) $(z \rightarrow \bar{x})(x \rightarrow y)$.

37. Доказать, что в каждой из следующих пар булевых функций одна из функций двойственна другой:

- (а) $xyz \oplus yz \oplus y \oplus 1, xyz \oplus xy \oplus xz \oplus x \oplus y \oplus 1$;
- (б) $xyz \oplus xz \oplus x \oplus y \oplus z, xy \oplus xz \oplus x$;
- (в) $xyz \oplus xy \oplus z, xyz \oplus xz \oplus yz$;
- (г) $xy \oplus yz \oplus x \oplus y \oplus z \oplus 1, xy \oplus yz \oplus y \oplus 1$;
- (д) $xyz \oplus xy \oplus x \oplus y, xyz \oplus xz \oplus yz \oplus x \oplus y \oplus z \oplus 1$;
- (е) $xy \oplus x \oplus y \oplus z \oplus 1, xy \oplus z$.

38. Доказать, что следующие булевы функции самодвойственны:

- (а) $x(z \rightarrow y) \vee (y \rightarrow z)$;
- (б) $\bar{x}y \oplus \bar{x}\bar{z} \oplus y\bar{z}$;
- (в) $xy \oplus xz \oplus yz \oplus y \oplus z$;
- (г) $xz \oplus (x \oplus z)(y \oplus 1)$;
- (д) $(x \rightarrow y)(z \rightarrow z) \vee (x \vee y)(z \rightarrow z)$;
- (е) $(\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee z)(\bar{x} \vee \bar{y} \vee \bar{z})$.

39. Выяснить, сравнимы ли два данных набора, и если сравнимы, то какой из них меньший:

- (а) (0, 1, 0), (1, 0, 0);
- (б) (1, 0, 0, 1), (1, 0, 1, 1);
- (в) (1, 0, 0), (1, 0, 1);
- (г) (0, 1, 0, 0), (0, 1, 0, 1);
- (д) (1, 0, 0, 1, 0), (1, 0, 1, 1, 0);
- (е) (0, 1, 0, 0, 1), (0, 1, 0, 1, 0).

40. Какие из следующих функций являются монотонными и почему?

- (а) $x \wedge y \vee x \wedge z \vee \bar{x} \wedge z$;
- (б) $x \rightarrow (x \rightarrow y)$;
- (в) $(x \vee y) \oplus (\bar{x} \vee \bar{y})$;
- (г) $(x \vee y) \leftrightarrow (x \wedge y)$;
- (д) $x \vee x \wedge y \vee x \wedge z$;
- (е) $x \wedge y \vee y \wedge z \vee x \wedge z$.

41. Найти все монотонные булевы функции от одного аргумента. Найти все монотонные булевы функции от двух аргументов.

42. Исследовать функции на принадлежность к классам C_0, C_1, C_S, C_M, C_L :

- (а) $f(x, y, z) = xy \vee z(x \rightarrow z) \vee yz$;

$$(б) f(x, y, z) = (xyz) \rightarrow z \vee yz;$$

$$(в) f(x, y, z) = (xy \vee z(x \vee z)) yz.$$

43. Подтвердить или опровергнуть приведенные утверждения. Обосновать (возможно на примерах):

(а) если f – линейная функция, то \bar{f} также является линейной функцией;

(б) если f_1 и f_2 – линейные функции, то при $f_1 \neq f_2$ их дизъюнкция всегда является нелинейной функцией;

(в) если f_1 и f_2 – линейные булевы функции, то их конъюнкция всегда является нелинейной функцией;

(г) если f – нелинейная булева функция, то конъюнкция этой функции и ее инверсии есть линейная функция;

(д) если f – нелинейная булева функция, то ее инверсия есть линейная функция;

(е) если f – нелинейная булева функция, то $f \oplus f$ является линейной функцией;

(ж) если f – нелинейная булева функция, то дизъюнкция этой функции и ее инверсии есть линейная функция;

(з) конъюнкция двух монотонных функций является монотонной функцией;

(и) существуют монотонные функции, инверсии которых представляют собой монотонные функции;

(к) функция $f_1 \oplus f_2$ всегда монотонна, если f_1 и f_2 – монотонные функции.

44. Выяснить, являются ли системы функций функционально полными. Почему? Являются ли системы функций минимальным базисом? Обоснуйте.

$$(а) \{x \vee y \rightarrow z, xy, 1\};$$

$$(б) \{x \vee y \rightarrow z, xy, 0\};$$

$$(в) \{x \vee y \rightarrow z, xy, \bar{x}\};$$

$$(г) \{xy \vee xz \vee yz, \bar{x}, 1\};$$

$$(д) \{xy \vee xz \vee yz, x \leftrightarrow y, x \oplus y\};$$

$$(е) \{x \oplus y \oplus z, xy, \bar{x}\};$$

$$(ж) \{xy \oplus z, (x \leftrightarrow y) \oplus z, 1\};$$

$$(з) \{xy \oplus xz \oplus yz, 1, 0\};$$

$$(и) \{x \oplus y \oplus z, xy \oplus xz \oplus yz, 0, 1\}.$$

45. Из полной системы булевых функций выделите всевозможные минимальные базисы.

(а) $\{(x \vee y)(\bar{x} \vee \bar{y}), xy \oplus z, (x \oplus y) \leftrightarrow z, xy \vee xz \vee yz\}$;

(б) $\{xyz, x \oplus y \oplus z, 1, x\bar{y}\}$;

(в) $\{(x \oplus y)z, 1, xy \vee \bar{y}z, x \oplus y \oplus z\}$;

(г) $\{x \rightarrow (y \rightarrow z), x \oplus y, 0, xy \oplus z\}$;

(д) $\{xy \vee z, x \leftrightarrow y, 0, 1\}$;

(е) $\{x \oplus y \oplus z, xy, 0, \bar{x}\}$.

2. ФОРМАЛЬНЫЕ ТЕОРИИ

2.1. Исчисление высказываний как формальная теория

Формальная теория строится следующим образом.

1. Определяется множество *формул*, или правильно построенных выражений, образующих язык теории. Это множество задается конструктивными средствами (как правило, индуктивным определением) и, следовательно, перечислимо. Обычно оно и разрешимо.

2. Выделяется подмножество формул, называемых *аксиомами* теории. Множество может быть и бесконечным, во всяком случае, оно должно быть разрешимо.

3. Задаются *правила вывода теории*. Правило вывода $R(F_1, \dots, F_n, G)$ – это вычисляемое отношение на множестве формул. Если формулы F_1, \dots, F_n, G находятся в отношении R , то формула G называется *непосредственно выводимой* из F_1, \dots, F_n по правилу R .

Правила вывода записываются в виде $\frac{F_1, \dots, F_n}{G}$ или в виде

$F_1, \dots, F_n \vdash G$.

Формулы F_1, \dots, F_n называются *посылками* правила R , или *гипотезами*, а G – *следствием*, или *заключением*.

В формальной теории существует два типа высказываний: *высказывания самой теории (теоремы)*, которые рассматриваются как чисто формальные объекты, определенные ранее, и *высказывания о теории* (о свойствах ее теорем, доказательств и т. д.), которые формулируются на языке, внешнем по отношению к теории, – метаязыке – и называются *метатеоремами*.

Пример

Если удалось построить вывод B из A_1, \dots, A_n , то высказывание « B выводима из A_1, \dots, A_n » является метатеоремой, ее можно рассматривать как дополнительное («произвольное») правило вывода, которое можно присоединить к исходным и использовать в дальнейшем.

Формула G называется *теоремой теории T* , если в T существует вывод G из пустого множества формул.

Формальная теория T называется *разрешимой*, если существует алгоритм, позволяющий отличить теорему от не теоремы.

Формальная теория T называется *непротиворечивой*, если в ней нельзя одновременно доказать формулу F и ее отрицание.

Формальная теория T называется *полной* относительно содержащейся в ней теории S , если каждое истинное высказывание S отображается в некоторую теорему теории T .

Исчисление высказываний

Исчисление высказываний как формальная теория определяется следующим образом.

1. *Алфавит исчисления высказываний* состоит из переменных высказываний (пропозициональных букв) A, B, C, \dots , знаков логических связок $\vee, \wedge, \neg, \rightarrow$ и скобок $(,)$.

Формулы:

- переменное высказывание является формулой;
- если A и B формулы, то $(A \vee B), (A \wedge B), (A \rightarrow B), \bar{A}$ – также формулы;
- других формул нет.

Внешние скобки у формул обычно принято опускать: например, вместо $(A \vee B)$ пишут $A \vee B$.

2. *Аксиомы* – тождественно-истинные высказывания, входящие в любую теорию в качестве законов.

Рассмотрим две системы аксиом. Первая непосредственно использует все логические связки.

Система аксиом I:

$$A 1.1 \quad A \rightarrow (B \rightarrow A);$$

$$A 1.2 \quad (A \rightarrow B) \rightarrow (A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C);$$

$$A 1.3 \quad (A \wedge B) \rightarrow A;$$

$$A 1.4 \quad (A \wedge B) \rightarrow B;$$

$$A 1.5 \quad A \rightarrow (B \rightarrow (A \wedge B));$$

$$A 1.6 \quad A \rightarrow (A \vee B);$$

$$A 1.7 \quad B \rightarrow (A \vee B);$$

$$A 1.8 \quad (A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C));$$

$$A 1.9 \quad (A \rightarrow B) \rightarrow ((A \rightarrow \neg B) \rightarrow \neg A);$$

$$A 1.10 \quad \neg\neg A \rightarrow A.$$

Другая система использует только две связки: \rightarrow и \neg . При этом сокращается алфавит исчисления (выбрасываются знаки \vee, \wedge) и, соответственно, определения формулы.

Система аксиом 2:

A 2.1 $A \rightarrow (B \rightarrow A)$;

A 2.2 $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B)) \rightarrow (A \rightarrow C)$;

A 2.3 $(\neg A \rightarrow \neg B) \rightarrow ((\neg A \rightarrow B) \rightarrow A)$.

Приведенные системы аксиом равносильны в том смысле, что порождают одно и то же множество формул.

3. Правила вывода.

Правило подстановки. Если выводимая формула F содержит некоторую переменную A (обозначим этот факт $F(A)$) и существует произвольная формула B , то формула $F(B)$, получающаяся заменой всех вхождений A на формулу B , также выводима в исчислении высказываний:

$$B: \frac{F(A)}{F(B)}.$$

Пример

Пусть даны формулы $F = A \wedge C \rightarrow A$ и $B = C \rightarrow \bar{A}$. Если выполнить подстановку формулы B в формулу F вместо формулы A , то получим новую формулу F' :

$$F': \frac{A \wedge C \rightarrow A}{(C \rightarrow \bar{A}) \wedge C \rightarrow (C \rightarrow \bar{A})}.$$

Если проверить значения двух формул F и F' по таблицам истинности, то мы получим тождество двух формул.

Правила заключения. При выводе формулы из множества аксиом и посылок используют два основных правила заключения: *modus ponens* (*m.p.*) и *modus tollens* (*m.t.*).

Правило Modus Ponens: если A и $A \rightarrow B$ есть выводимые формулы, то B также выводимая формула, т. е.

$$\frac{A, A \rightarrow B}{B}.$$

Формальная запись высказывания (умозаключения, рассуждения) в виде формулы имеет вид $(A \cdot (A \rightarrow B)) \rightarrow B$. Проверка правильности рассуждения проводится по таблице истинности или при помощи эквивалентных преобразований:

$$\begin{aligned} (((A \rightarrow B) \wedge A) \rightarrow B) &= \overline{\overline{((\bar{A} \vee B) \wedge A)} \vee B} = \\ &= \overline{((A \wedge \bar{B}) \vee \bar{A}) \vee B} = (\bar{B} \vee \bar{A} \vee B) = 1. \end{aligned}$$

Правильное рассуждение – рассуждение, формула которого является тавтологией.

Пример

Суждение: «Сумма внутренних углов многоугольника равна 180° . Если сумма внутренних углов многоугольника равна 180° , то многоугольник есть треугольник. Следовательно, дан треугольник».

Введем высказывания:

$A :=$ «Сумма внутренних углов многоугольника равна 180° » и

$B :=$ «Многоугольник есть треугольник».

Тогда формальная запись суждения в виде формулы $\frac{A, A \rightarrow B}{B}$.

Пример

Дано суждение: «Если лекция скучная, то студент спит. Лекция скучная». Необходимо сделать вывод.

Введем высказывания:

$A :=$ «Лекция скучная» и

$B :=$ «Студент спит».

Тогда формальная запись суждения в виде формулы $A \rightarrow B, A$.

Применяя правило *modus ponens* $\frac{A, A \rightarrow B}{B}$ можно сделать вывод, что студент спит (B).

Правило Modus Tollens – рассуждение от противного (от лат. *modus tollendo tollens*, означает «путь исключения исключений»): если формулы \bar{A} и $A \rightarrow B$ есть выводимые формулы, то \bar{B} – также выводимая формула, т. е.

$$\frac{A, A \rightarrow \bar{B}}{\bar{A}}$$

Пример

Суждение: «Дан не треугольник. Если сумма внутренних углов многоугольника равна 180° , то многоугольник есть треугольник. Следовательно, сумма внутренних углов многоугольника не равна 180° ».

Введем высказывания:

$A :=$ «Сумма внутренних углов многоугольника равна 180° » и

$B :=$ «Многоугольник есть треугольник».

Тогда формальная запись суждения в виде формулы $\frac{A, A \rightarrow \bar{B}}{\bar{A}}$.

Пример

Дано суждение: «Если лекция скучная, то студент спит. Студент не спит». Необходимо сделать вывод.

Введем высказывания:

$A :=$ «Лекция скучная» и

$B :=$ «Студент спит».

Тогда формальная запись суждения в виде формулы $A \rightarrow B, \bar{B}$.

Применяя правило *modus tollens* $\frac{A, A \rightarrow \bar{B}}{\bar{A}}$, можно сделать вывод, что лекция не скучная (\bar{A}).

Рассмотрим *примеры вывода в исчислении высказываний*.

Пример

Покажем, что формула $A \rightarrow A$ выводима из системы аксиом 2:
 $\vdash A \rightarrow A$.

Решение

1. $(A \rightarrow ((A \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))$ – на данном шаге осуществлена подстановка в аксиому A2.2 ($A \rightarrow A$) вместо B и A вместо C .

2. $A \rightarrow ((A \rightarrow A) \rightarrow A)$ – произведена подстановка в аксиому A2.1 ($A \rightarrow A$) вместо B .

3. $(A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$ – по правилу заключения *modus ponens* из шагов 1 и 2.

4. $A \rightarrow (A \rightarrow A)$ – произведена подстановка в аксиому A2.1 A вместо B .

5. $A \rightarrow A$ – по правилу заключения *modus ponens* из шагов 3 и 4.

Пример

Докажем, что $A \vdash B \rightarrow A$.

Пусть A выводима. Тогда из A и аксиомы A2.1 по правилу заключения получаем $\frac{A, A \rightarrow (B \rightarrow A)}{B \rightarrow A}$, что и доказывает искомую выводимость.

Полученную выводимость $A \vdash B \rightarrow A$ вместе с правилом подстановки можно рассматривать, как правило $\frac{A}{B \rightarrow A}$: «Если формула A выводима, то выводима и формула $B \rightarrow A$, где B – любая формула».

Для построения выводов в исчислении высказываний применяется метод дедукции, основанный на выявлении общих закономерностей в процессе построения доказательств вывода формул.

Теорема дедукции. Если $F_1, \dots, F_{n-1}, F_n \vdash G$, то $F_1, \dots, F_{n-1} \vdash F_n \rightarrow G$. В частности, если $F \vdash G$, то $\vdash F \rightarrow G$.

Теорема дедукции $F_1, F_2, \dots, F_n \vdash B$ равносильна доказательству $\vdash (F_1 \wedge F_2 \wedge \dots \wedge F_n \rightarrow B)$. Если каждая $F_i = \text{И}$, то $F_1 \wedge F_2 \wedge \dots \wedge F_n = \text{И}$, а если $(F_1 \wedge F_2 \wedge \dots \wedge F_n \rightarrow B) = \text{И}$, то $B = \text{И}$.

Следовательно, при истинности всех посылок и истинности импликации (по правилу *m.p.*) заключение всегда будет истинным.

Используя правила эквивалентных преобразований алгебры высказываний, можно показать дедуктивный характер вывода заключения:

- 1) $\vdash (F_1 \wedge F_2 \wedge \dots \wedge F_n \rightarrow B)$;
- 2) $\vdash ((\overline{F_1 \wedge F_2 \wedge \dots \wedge F_n}) \vee B)$;
- 3) $\vdash (\overline{F_1} \vee \overline{F_2} \vee \dots \vee \overline{F_n} \vee B)$;
- 4) $\vdash (\overline{F_1} \vee \overline{F_2} \vee \dots \vee \overline{F_{n-1}} \vee (F_n \rightarrow B))$;
- 5) $\vdash (\overline{F_1} \vee \overline{F_2} \vee \dots \vee (F_{n-1} \rightarrow (F_n \rightarrow B)))$;
- 6) $\vdash (\overline{F_1} \vee (F_2 \rightarrow \dots \rightarrow (F_{n-1} \rightarrow (F_n \rightarrow B))))$;
- 7) $\vdash \vdash (F_1 \rightarrow (F_2 \rightarrow \dots \rightarrow (F_{n-1} \rightarrow (F_n \rightarrow B))))$;

Так формируется система дедуктивного вывода от посылок до заключения.

Пример

Дано суждение: «Всякое общественно опасное деяние наказуемо. Преступление есть общественно опасное деяние. Дача взятки – преступление. Следовательно, дача взятки наказуема?».

Введем высказывания:

$A :=$ «Деяние – общественно опасное»;

$B :=$ «Деяние наказуемо»;

$C :=$ «Деяние – преступление»;

$D :=$ «Дача взятки».

Тогда формальная запись суждения в виде формулы

$$\frac{A \rightarrow B, C \rightarrow A, D \rightarrow C}{D \rightarrow B}.$$

Построим последовательность вывода от посылок через промежуточные результаты к заключению:

- 1) $F_1 = A \rightarrow B$ – посылка;
- 2) $F_2 = C \rightarrow A$ – посылка;
- 3) $F_3 = D \rightarrow C$ – посылка;
- 4) $F_4 = C \rightarrow B$ – заключение по правилу подстановки;
- 5) $F_5 = D \rightarrow B$ – заключение по правилу подстановки.

Таким образом доказана истинность заключения ($D \rightarrow B$). Следовательно, дача взятки наказуема.

2.2. Исчисление предикатов как формальная теория

Логика предикатов

Предикат (лат. *praedicatum* – «логическое сказуемое») – это функция одного или нескольких аргументов с булевыми значениями «истина» и «ложь».

Введем строгое определение. *Предикатом* называется отображение прямого произведения заданных множеств в множество значений истинности $P: M_1 \times M_2 \times \dots \times M_n \rightarrow \{0, 1\}$, где M_1, M_2, \dots, M_n – заданные произвольные множества.

Если $P(x_1, x_2, \dots, x_n)$, то аргумент предиката $x_i \in M$ является *предметной переменной*. Конкретные значения аргументов предиката называют *предметными постоянными*.

Например, если на множестве имен индивидов, университетов и специальностей задать предикаты: $P_1(x) := \langle x - \text{студент} \rangle$, $P_2(x, \text{СФУ}) := \langle \text{студент } x \text{ университета СФУ} \rangle$, где x есть предметная переменная, а СФУ – предметная постоянная, – то высказываниями будут $P_1(\text{Петров}) = 1$, $P_2(\text{Петров}, \text{СФУ}) = 1$.

При ограничении области определения предметных переменных вводят операторы, которые называют *кванторами*.

Суждение, в котором утверждается или отрицается наличие каких-либо признаков или отношений у части предметных переменных области определения, называют *частным суждением*. Как правило, эти суждения на естественном языке отражают словами «один», «несколько», «часть» и т. п. Для формализации таких суждений используют логическую операцию, ограничивающую область определения предиката – оператор, называемый *квантор существования*, который

обозначают $\exists x$. Предикат записывают после квантора существования в круглых скобках – $\exists x (P_n(x))$.

Эта запись читается: «существуют такие элементы x , что $P_n(x)$ истинно (или ложно)».

Если частное суждение распространяется на несколько предметных переменных, то перед предикатом записывают все предметные переменные, по которым есть частное суждение, т. е. $\exists x, \exists y, \exists z, \dots, (P_n(x, y, z, \dots))$.

Суждение, в котором утверждается или отрицается наличие каких-либо признаков или отношений для всех предметных переменных области определения, называют *общими суждениями*. Такие суждения в естественном языке выражают словами «все», «каждый», «любой» и т. п. Для формализации этих суждений используют логическую операцию над всей областью определения предиката – оператор, называемый *квантор всеобщности (общности)*, обозначаемый $\forall x$. Правила записи предикатов с квантором всеобщности аналогичны правилам записи с квантором существования.

Пример

$\forall x (P_n(x))$ – «для всех x истинно (или ложно) значение $P_n(x)$ ».

Существуют предикаты, для которых область определения по различным предметным переменным ограничивают различными кванторами.

Предметную переменную предиката, если по меньшей мере одно ее вхождение связано квантором, называют *связанной переменной*. Предметную переменную предиката, если по меньшей мере одно ее вхождение в формулу свободно от квантора, называют *свободной переменной*.

Пример

$\exists y (A(x, y))$:= «для всех целых чисел x существуют меньшие числа y ». В этом примере x – свободная, а y – связанная переменные.

Понятие свободной переменной подобно понятию глобальной переменной, т. е. переменной вне текущей процедуры, а понятие связанной переменной подобно понятию локальной переменной для текущей процедуры.

Если предикат содержит один аргумент, то задан *одноместный предикат*, если она содержит n аргументов, то *n -местный предикат*.

Одноместный предикат, как правило, описывает наличие какого-либо признака у предмета, а n -местный предикат – наличие отношений между n предметами.

Пример

$P(x) :=$ « x – студент» – одноместный предикат.

$P(x, y) :=$ «Студент x находится в аудитории y » – двуместный предикат.

Следует обратить внимание, что когда все предметные переменные замещены предметными постоянными, тогда предикат превращается в высказывание.

Простейшими логическими операциями над предикатами являются отрицание, конъюнкция, дизъюнкция, импликация и эквивалентность. Рассмотрим запись утверждений на языке логики предикатов.

Пример

Если $P_1(x) :=$ «Выдающийся программист» и $P_2(x) :=$ «Талантливый поэт», то получаем следующие формулы: $\exists x(P_1(x) \wedge P_2(x)) :=$ «Существуют выдающиеся программисты и талантливые поэты» и $\exists x(\overline{P_1(x) \wedge P_2(x)}) :=$ «существуют лица, являющиеся выдающимися программистами и не являющиеся талантливыми поэтами».

Рассмотренные логические операции позволяют формализовать с помощью термов, предикатов и кванторов внутреннюю структуру предложения и формировать сложные суждения.

Пример

Суждение: «Саша – мальчик, у которого нет машины. Таня – девочка, которая любит мальчиков, имеющих машины. Следовательно, Таня не любит Сашу».

В этом суждении два одноместных предиката $P_1(x) :=$ «Быть мальчиком», $P_2(x) :=$ «Быть девочкой», и два двуместных $P_3(x, y) :=$ « x любит y », $P_4(x, y) :=$ « x имеет y », три высказывания $P_1(a) :=$ «Саша – мальчик», $P_2(b) :=$ «Таня – девочка» и $\neg P_4(a, c) :=$ «Саша не имеет машины».

Формула сложного суждения должна быть записана так:

$$\frac{P_1(a), P_2(b), \overline{P_4(a, c)}, \exists x(P_2(x) \wedge \forall y(P_1(y) \wedge P_4(y, c) \rightarrow P_3(x, y)))}{P_2(b) \wedge \overline{P_3(b, a)}}$$

Приведенные примеры позволяют сформулировать некоторые правила записи сложных суждений в логике предикатов:

1) каждое вхождение логической связки «отрицание» относится к формуле, следующей непосредственно за логической связкой справа;

2) каждое вхождение логической связки « \wedge » после расстановки скобок связывает формулы, непосредственно окружающие логическую связку;

3) каждое вхождение логической связки « \vee » после расстановки скобок связывает формулы, непосредственно окружающие эту связку;

4) логические связки по силе и значимости могут быть упорядочены следующим образом: \neg ; \wedge ; \vee ; \rightarrow ; \leftrightarrow ;

5) за квантором общности чаще всего следует логическая связка импликации, а за квантором существования – конъюнкции;

6) если формула содержит подформулу, то внутренняя формула не должна содержать кванторов, связывающих ту же переменную, что и квантор формулы;

7) значения всех предметных переменных и постоянных должны принадлежать одной области определения предиката или функции;

8) если в одной формуле есть кванторы общности и существования, то при формализации суждений следует стремиться поставить квантор существования слева всей формулы.

Исчисление предикатов как формальная теория

Исчисление предикатов как формальная теория определяется следующим образом.

1. *Алфавит исчисления предикатов*, состоящий из:

- предметных переменных $x, y, \dots, x_1, y_1, \dots$;
- предметных констант $a, b, \dots, a_1, b_1, \dots$;
- предикатных переменных P, Q, \dots ;
- функциональных букв (функторов) f, g, \dots ;
- знаков логических связок $\wedge, \vee, \neg, \rightarrow$;
- кванторов \forall, \exists ;
- служебных символов $(,)$.

2. *Аксиомы*, которые совпадают с аксиомами исчисления высказываний А1.1–А1.10, за исключением двух добавленных аксиом с кванторами:

$$P1. \forall xF(x) \rightarrow F(y);$$

$$P2. F(y) \rightarrow \exists xF(x).$$

3. Правила вывода исчисления предикатов – такие же, как для исчисления высказываний. К ним добавлены правила вывода для работы с кванторами:

- $\frac{F \rightarrow G(x)}{F \rightarrow \forall x F(x)}$ – правило обобщения (введения квантора всеобщности \forall);
- $\frac{G(x) \rightarrow F}{\exists x G(x) \rightarrow F(x)}$ – правило обобщения (введения квантора существования \exists).

Исчисление предикатов, в котором кванторы могут связывать только предметные переменные, но не могут связывать функторы или предикаты, называется *исчислением первого порядка*.

Исчисления, в которых кванторы могут связывать не только предметные переменные, но и функторы, предикаты или иные множества объектов, называются *исчислениями высшего порядка*.

Все операции над значениями истинности можно обобщить до операций над множествами предикатов над заданным множеством M .

Формула в исчислении предикатов определяется следующим образом:

- каждая предикатная переменная есть формула;
- если $P(\dots, \dots, \dots)$ – n -местная предикатная переменная, то $P(x_1, \dots, x_n)$ – формула, в которой все предметные переменные x_1, \dots, x_n – свободны;
- если F_1 и F_2 – формулы, то $\overline{F_1}$; $\overline{F_2}$; $F_1 \wedge F_2$; $F_1 \vee F_2$; $F_1 \rightarrow F_2$ и $F_1 \leftrightarrow F_2$ – также формулы, в которых все вхождения переменных, свободные (связанные) в формулах F_1, F_2 , являются свободными (связанными) и в новых формулах;
- если $F(x)$ – формула, содержащая свободные вхождения переменной x , то $\forall x F(x)$ и $\exists x F(x)$ – формулы, в которых переменная x – связанная, а вхождения всех остальных переменных в F остаются свободными;
- никаких других формул в логике предикатов нет.

В формулах $\forall x A$ и $\exists x A$ подформула A называется *областью действия квантора по x* .

Исчисление предикатов, которое не содержит предметных констант, функторов, предикатов и собственных аксиом, называется *чистым исчислением предикатов*.

Исчисление предикатов, которое содержит предметные константы и/или функторы и/или предикаты и связывающие их собственные аксиомы, называются *прикладным исчислением предикатов*.

Эквивалентные соотношения для формул логики предикатов

Формулы называются *эквивалентными*, если при любых подстановках констант они принимают одинаковые значения. В частности, все тождественно истинные формулы (и все ложные формулы) эквивалентны.

Отметим, что если формулы F_1 и F_2 эквивалентны в соответствии с этим определением, то формула $F_1 \sim F_2$ тождественно истинна.

Если формула алгебры предикатов F имеет вхождением подформулу F_i , т. е. $F(t_1; t_2; \dots; F_i; \dots)$, для которой существует эквивалентная ей подформула F_j т. е. $F_i = F_j$, то возможна подстановка всюду в формулу F вместо формулы F_i подформулы F_j без нарушения истинности формулы, т. е.

$$F(t_1; t_2; \dots; F_i; \dots) = F(t_1; t_2; \dots; F_j; \dots).$$

Если в законах логики высказываний вместо имеющихся пропозициональных переменных всюду подставить предикаты так, чтобы вместо одной и той же пропозициональной переменной стоял один и тот же предикат, то получаем закон логики предикатов.

Рассмотрим основные *законы эквивалентных преобразований логики предикатов*.

1. Коммутативность (*только для одноименных кванторов*):

а) $\forall x \forall y (F(x, y)) = \forall y \forall x (F(x, y))$;

б) $\exists x \exists y (F(x, y)) = \exists y \exists x (F(x, y))$.

2. Дистрибутивность:

а) $\forall x (F_1(x) \wedge \forall x (F_2(x))) = \forall x (F_1(x) \wedge F_2(x))$;

б) $\exists x (F_1(x) \vee \exists x (F_2(x))) = \exists x (F_1(x) \vee F_2(x))$.

3. Идемпотентность:

а) $\mathfrak{R}x(F(x)) \vee \mathfrak{R}x(F(x)) = \mathfrak{R}x(F(x))$;

б) $\mathfrak{R}x(F(x)) \wedge \mathfrak{R}x(F(x)) = \mathfrak{R}x(F(x))$, где $\mathfrak{R} \in \{\forall; \exists\}$.

4. Закон исключения третьего:

$$\mathfrak{R}x(F(x)) \vee \overline{\mathfrak{R}x(F(x))} = 1, \text{ где } \mathfrak{R} \in \{\forall; \exists\}.$$

5. Закон противоречия:

$$\mathfrak{R}x(F(x)) \wedge \overline{\mathfrak{R}x(F(x))} = 0, \text{ где } \mathfrak{R} \in \{\forall; \exists\}.$$

6. Законы де Моргана:

а) $\forall x(\overline{F(x)}) = \overline{\exists x(F(x))}$;

б) $\exists x(\overline{F(x)}) = \overline{\forall x(F(x))}$;

7. Двойное отрицание:

$$\overline{\overline{\mathfrak{R}x(F(x))}} = \mathfrak{R}x(F(x)), \text{ где } \mathfrak{R} \in \{\forall; \exists\}.$$

8. Свойства констант:

а) $\mathfrak{R}x(F(x)) \vee 1 = 1$;

б) $\mathfrak{R}x(F(x)) \vee 0 = \mathfrak{R}x(F(x))$;

в) $\mathfrak{R}x(F(x)) \wedge 0 = 0$;

г) $\mathfrak{R}x(F(x)) \wedge 1 = \mathfrak{R}x(F(x))$, где $\mathfrak{R} \in \{\forall; \exists\}$.

Рассмотрим на примере применение эквивалентных преобразований логики предикатов для упрощения формул.

Пример

Упростить формулу

$$F = \forall x(P_1(x) \rightarrow \neg P_2(x)) \rightarrow \neg(\exists x(P_1(x)) \wedge \forall x(P_2(x))).$$

Решение:

1) удалить логическую связку « \rightarrow »:

$$F = \neg(\forall x(\neg P_1(x) \vee \neg P_2(x))) \vee \neg(\exists x(P_1(x)) \wedge \forall x(P_2(x)));$$

2) выполнить операцию отрицания формулы:

$$F = \exists x(\neg(\neg P_1(x) \vee \neg P_2(x))) \vee \forall x(\neg P_1(x) \vee \exists x(\neg P_2(x)));$$

3) выполнить операцию отрицания формулы:

$$F = \exists x(P_1(x) \wedge P_2(x)) \vee \forall x(\neg P_1(x) \vee \exists x(\neg P_2(x)));$$

4) применить закон дистрибутивности по квантору $\exists x$:

$$F = \exists x((P_1(x) \wedge P_2(x)) \vee \neg P_2(x)) \vee \forall x(\neg P_1(x));$$

5) применить закон дистрибутивности к формуле:

$$F = \exists x((P_1(x) \vee \neg P_2(x)) \wedge (P_2(x) \vee \neg P_2(x))) \vee \forall x(\neg P_1(x));$$

6) применить закон исключенного третьего и свойство констант для логической связки « \wedge »:

$$F = \exists x(P_1(x) \vee \neg P_2(x)) \vee \forall x(\neg P_1(x));$$

7) применить закон де Моргана:

$$F = \exists x(P_1(x) \vee \neg P_2(x)) \vee \neg \exists x(P_1(x));$$

8) применить закон дистрибутивности по квантору $\exists x$:

$$F = \exists x(P_1(x)) \vee \exists x(\neg P_2(x)) \vee \neg \exists x(P_1(x));$$

9) применить закон исключенного третьего:

$$F = \exists x(\neg P_2(x)) \vee 1;$$

10) применить свойство констант для логической связки « \vee »:

$$F = 1,$$

т. е. формула

$$F = \forall x(P_1(x) \rightarrow \neg P_2(x)) \rightarrow \neg(\exists x(P_1(x)) \wedge \forall x(P_2(x)))$$

является тождественно истиной.

Предваренная нормальная форма в логике предикатов

Для облегчения анализа сложных суждений формулы алгебры предикатов рекомендуется приводить к *предваренной нормальной форме (ПНФ)*. Суть ПНФ сводится к разделению формулы на две части: кванторную и безкванторную. Для этого все кванторы формулы выносят влево, используя законы и правила алгебры предикатов.

В результате этих алгебраических преобразований может быть получена формула вида $\mathfrak{R}_1 \mathfrak{R}_2 \dots \mathfrak{R}_n (M)$, где $\mathfrak{R} \in \{\forall; \exists\}$, а M – *матрица формулы*, кванторную часть формулы $\mathfrak{R}_1 \mathfrak{R}_2 \dots \mathfrak{R}_n$ называют *префиксом ПНФ*.

Пример

$$F = \exists x \forall y ((P_1^2(x, y) \vee \overline{P_2(x)} \wedge P_3(y))) -$$

формула, приведенная к ПНФ;

$$F = \forall y (P_1^2(x, y) \vee \exists x \overline{P_2(x)}) \wedge \exists y (P_3(y)) -$$

формула, не приведенная к ПНФ.

$$\forall x (P_1(x)) \wedge \forall x P_2(x) = \forall x (P_1(x) \wedge P_2(x)) -$$

слева от знака равенства формула, неприведенная к ПНФ, а справа – равносильная ей формула, но приведенная к ПНФ.

В последующем матрицу формулы преобразуют к виду КНФ, что облегчает механизм по принципу резолюции.

Рассмотрим алгоритм приведения формулы к ПНФ.

Шаг 1. Перейти в систему булевых операций.

Шаг 2. Продвинуть отрицание до элементарной формулы.

Шаг 3. Переименовать связанные переменные по правилу: «найти самое левое вхождение предметной переменной такое, что это вхождение связано некоторым квантором, но существует еще одно вхождение этой же переменной; затем сделать замену связанного вхождения на вхождение новой переменной», операцию повторять пока возможна замена связанных переменных.

Шаг 4. Вынести кванторы влево по законам алгебры логики.

Шаг 5. Преобразовать бескванторную матрицу к виду КНФ.

Пример

$$F = (\forall x(P_1(x) \rightarrow \forall y(P_2(y) \rightarrow P_3(z)))) \wedge (\forall y(\overline{P_4(x, y) \rightarrow P_5(z)})).$$

Привести формулу к виду ПНФ:

1) удалить логические связки « \rightarrow »:

$$F = (\forall x(\overline{P_1(x)} \vee \forall y(\overline{P_2(y)} \vee P_3(z)))) \wedge (\forall y(\overline{P_4(x, y)} \vee P_5(z)));$$

2) применить закон де Моргана $\overline{\forall x(F(x))} = \exists x(\overline{F(x)})$:

$$F = (\forall x(\overline{P_1(x)} \vee \forall y(\overline{P_2(y)} \vee P_3(z)))) \wedge (\exists y(\overline{P_4(x, y)} \vee P_5(z)));$$

3) применить закон де Моргана $\overline{(F_1 \vee F_2)} = (\overline{F_1} \wedge \overline{F_2})$:

$$F = (\forall x(\overline{P_1(x)} \vee \forall y(\overline{P_2(y)} \vee P_3(z)))) \wedge (\exists y(P_4(x, y) \wedge \overline{P_5(z)}));$$

4) переименовать связанную переменную $x = w$:

$$F = (\forall w(\overline{P_1(w)} \vee \forall y(\overline{P_2(y)} \vee P_3(z)))) \wedge (\exists y(P_4(x, y) \wedge \overline{P_5(z)}));$$

5) переименовать связанную переменную $y = v$:

$$F = (\forall w(\overline{P_1(w)} \vee \forall v(\overline{P_2(v)} \vee P_3(z)))) \wedge (\exists y(P_4(x, y) \wedge \overline{P_5(z)}));$$

6) вынести квантор $\forall v$ влево:

$$F = \forall w \forall v (\overline{P_1(w)} \vee \overline{P_2(v)} \vee P_3(z)) \wedge (\exists y(P_4(x, y) \wedge \overline{P_5(z)}));$$

7) вынести квантор $\exists y$ влево:

$$F = \forall w \forall v \exists y (\overline{P_1(w)} \vee \overline{P_2(v)} \vee P_3(z)) \wedge P_4(x, y) \wedge \overline{P_5(z)}.$$

Матрица ПНФ содержит три элементарных дизъюнкта:

$$K = \{\overline{P_1(w)} \vee \overline{P_2(v)} \vee P_3(z); P_4(x, y); \overline{P_5(z)}\}.$$

2.3. Автоматическое доказательство теорем. Принцип резолюций

Пусть имеется множество формул $T: F_1, F_2, \dots, F_n$ и формула H . Автоматическим доказательством теоремы H называют алгоритм, который проверяет вывод $F_1, F_2, \dots, F_n \vdash H$.

В общем случае такой алгоритм построить нельзя. Но для некоторых частных случаев такие алгоритмы существуют. Наиболее эффективным применяемым методом является метод резолюций.

Метод (принцип) резолюций был предложен Ж. Эрбраном в 1930 г., а впервые реализован на ЭВМ Дж. Робинсоном в 1963 г.

В основе метода резолюций лежит идея доказательства от противного или процедура поиска опровержения, т. е. вместо доказательства тождественной истинности формулы доказывається, что отрицание формулы противоречиво. В основе алгоритма лежит теорема Эрбрана.

Теорема Эрбрана. Чтобы доказать теорему, имеющую вид $T: F_1, F_2, \dots, F_n \vdash H$, в исчислении предикатов достаточно доказать логическую противоречивость (тождественную ложность) формулы $F_1 \wedge F_2 \wedge \dots \wedge F_n \wedge \overline{H}$.

Формула $F_1 \wedge F_2 \wedge \dots \wedge F_n \wedge \overline{H} = 0$ при ложности хотя бы одной из подформул F_i или \overline{H} .

Теорема о резолюции. Пусть имеется некоторая формула, представленная в конъюнктивной нормальной форме:

$$F = C_1 \wedge C_2 \wedge \dots \wedge (p \vee l_i) \wedge \dots \wedge (\overline{p} \vee l_i) \wedge \dots \wedge C_k,$$

и некоторая формула $E = F \wedge (l_i \vee \overline{l_i})$. Тогда, если E – противоречива, то и F – противоречива. Другими словами, для противоречивости F достаточно показать противоречивость E .

Предположим, что E противоречива. Тогда либо F противоречива (и теорема доказана), либо $(l_i \vee l_i)$ противоречива. Покажем, что в последнем случае F также противоречива: $(l_i \vee l_i) \equiv false \Rightarrow l_i = false$ и $l_j = false$. Подставляя $l_i = false$ и $l_j = false$ в выражение F , путем равносильных преобразований получим:

$$F = C_1 \wedge C_2 \wedge \dots \wedge p \wedge \dots \wedge \bar{p} \dots \wedge C_k \equiv false.$$

Теорема доказана.

Член $(l_i \vee l_i)$ называется *резольвентой*, а $C_i = p \vee l_i$, $C_j = \bar{p} \vee l_j$ – *родительскими предложениями*. Операция нахождения резольвенты называется *резольвцией*.

Следствие из теоремы. Чтобы доказать противоречивость формулы, достаточно найти ее пустую резольвенту.

Рассмотрим алгоритм вывода $F_1, F_2, \dots, F_n \vdash H$ по принципу резольвции в исчислении высказываний.

Шаг 1. Принять отрицание заключения, т. е. \bar{H} .

Шаг 2. Привести все формулы посылок F_1, F_2, \dots, F_n и \bar{H} к КНФ.

Шаг 3. Выписать множество элементарных дизъюнкций F_1, F_2, \dots, F_n и \bar{H} :

$$K = \{C_1, C_2, \dots, C_k\}.$$

Шаг 4. Выполнить анализ пар множества K по правилу: «если существуют дизъюнкты C_i и C_j , один из которых содержит переменную A , а другой – контрарную переменную \bar{A} , то соединить эту пару логической связкой дизъюнкции $C_i \vee C_j$ и сформировать новый дизъюнкт – резольвенту, исключив контрарные переменные A и \bar{A} ».

Шаг 5. Если в результате соединения дизъюнктов, содержащих контрарные переменные, будет получена пустая резольвента, то конец (доказательство подтвердило противоречие), в противном случае включить резольвенту в множество дизъюнктов K и перейти к шагу 4.

Пример

Покажем противоречивость формулы исчисления высказываний:

$$F = \bar{s} \wedge q \wedge (p \vee \bar{q}) \wedge (\bar{q} \vee s).$$

Выпишем родительские предложения:

$$C_1 = \bar{s}.$$

$$C_2 = q.$$

$$C_3 = p \vee \bar{q}.$$

$$C_4 = \bar{q} \vee s.$$

По C_1 и C_4 построим резольвенту $C_5 = \bar{q}$.

Из C_2 и C_5 получаем пустую резольвенту C_6 .

Следовательно, формула $F = \bar{s} \wedge q \wedge (p \vee \bar{q}) \wedge (\bar{q} \vee s)$ противоречива.

Пример

Задано следующее описание: «В совершении преступления подозреваются четверо человек: Адамс, Браун, Симпсон, Джонсон. Установлено следующее:

- если Адамс или Браун виновны, то и подозреваемый Симпсон виновен;
- если Адамс виновен, то по крайней мере, один из двух – Браун или Симпсон – тоже виновен;
- если Симпсон виновен в ограблении, то и Джонсон виновен;
- если Адамс виновен, то Джонсон виновен.

Виновен ли Джонсон?»

Решение

Введем следующие высказывания:

A := « A виновен».

B := « B виновен».

C := « C виновен».

D := « D виновен».

Запишем четыре известных факта и ответ на вопрос в виде логики предикатов:

$$1. F_1 : A \vee B \rightarrow C.$$

$$2. F_2 : A \rightarrow B \vee C.$$

$$3. F_3 : C \rightarrow D.$$

$$4. F_4 : \bar{A} \rightarrow D.$$

$$5. \text{Следствие} - G : D.$$

По теореме Эрбрана построим формулу $F_1 \wedge F_2 \wedge F_3 \wedge F_4 \wedge \bar{G}$:

$$F = (A \vee B \rightarrow C) \wedge (A \rightarrow B \vee C) \wedge (C \rightarrow D) \wedge (\bar{A} \rightarrow D) \wedge \bar{D}.$$

Приведем дизъюнкты к КНФ:

$$1. A \vee B \rightarrow C \equiv (\overline{A \vee B}) \vee C \equiv \overline{A} \wedge \overline{B} \vee C \equiv (\overline{A} \vee C) \wedge (\overline{B} \vee C).$$

$$2. A \rightarrow B \vee C \equiv \overline{A} \vee B \vee C.$$

$$3. C \rightarrow D \equiv \overline{C} \vee D.$$

$$4. \overline{A} \rightarrow D \equiv A \vee D.$$

Получили массив родительских предложений:

$$C_1 = \overline{A} \vee C.$$

$$C_2 = \overline{B} \vee C.$$

$$C_3 = \overline{A} \vee B \vee C.$$

$$C_4 = \overline{C} \vee D.$$

$$C_5 = A \vee D.$$

$$C_6 = \overline{D}.$$

По теореме о резолюции построим резольвенты:

$$C_7 = \overline{A} \vee D \text{ для } C_1 \text{ и } C_4.$$

$$C_8 = A \text{ для } C_5 \text{ и } C_7.$$

C_8 – пустая резольвента.

Следовательно, формула F противоречива. Результат: «Подозреваемый Джонсон – виновен». Дерево решений данной задачи представлено на рис. 6.

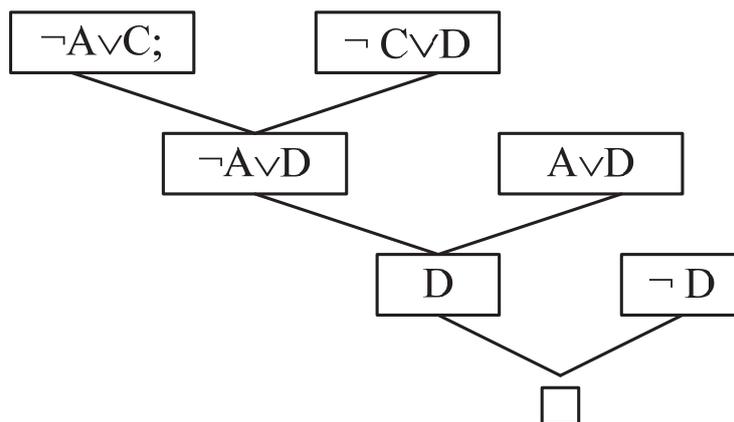


Рис. 6. Дерево решений

Принцип резолюции в исчислении предикатов

Рассмотрим механизм вывода по принципу резолюции в исчислении предикатов.

Шаг 1. Приведение формулы к предваренной нормальной форме. Для этого все кванторы выносятся в префиксную часть с сохранением их порядка и области действия.

Шаг 2. Удаление кванторов существования. Квантор существования можно отбросить, заменив все вхождения связываемой переменной:

- на константу, если квантору \exists не предшествует квантор всеобщности \forall : $(\exists x) A(x)$ заменяется на $A(c)$, где c – константа – значение переменной, для которого утверждение A имеет место;

- на функцию Сколема, если квантору существования предшествует квантор всеобщности, то при отбрасывании квантора \exists все вхождения связываемой переменной $(\forall y)(\exists x) A(x)$ заменяется на $(\forall y) A(f(y))$, где $f(y)$ – функция Сколема, которую можно интерпретировать как псевдоконстанту, имеющую тот же смысл, что и c , с той разницей, что для каждого значения переменной y имеется своя константа, для которой имеет место утверждение A . Соответственно, если квантору существования предшествует несколько кванторов всеобщности, функция Сколема будет иметь несколько переменных.

Шаг 3. Кванторы всеобщности \forall отбрасываются. Таким образом, получим бескванторную формулу.

Шаг 4. Приведение формулы путем равносильных преобразований к базису \wedge, \vee, \neg .

Шаг 5. Получение КНФ.

Шаг 6. Все члены КНФ представляются в виде массива родительских предложений, куда потом помещаются и резольвенты, причем:

- каждое предложение хранится в единственном экземпляре;
- допускается применение закона поглощения (менее общие предложения отбрасываются).

Шаг 7. Операция резолюции выполняется до тех пор, пока не получится пустая резольвента.

Основное ограничение метода резолюций состоит в том, что, поскольку все доказательства ведутся по условию *достаточности*, этот метод гарантирует получение пустой резольвенты только в том случае, если исходная формула действительно противоречива.

Пример

Используя метод резолюций в логике предикатов, доказать логичность следующих рассуждений: «Таможенники обыскивают каждого, кто въезжает в страну, кроме высокопоставленных лиц. Некоторые

люди, способствующие провозу запрещенных товаров, въезжали в страну и были обысканы исключительно людьми, способствующими провозу запрещенных товаров. Никто из высокопоставленных лиц не способствовал провозу запрещенных товаров. *Заключение*: некоторые из таможенников способствовали провозу запрещенных товаров».

Решение

Введем следующие предикаты:

$E(x)$ – « x – любой человек, въезжающий в страну»;

$V(x)$ – « x – высокопоставленное лицо»;

$S(x, y)$ – « y обыскивает x »;

$C(x)$ – « x является таможенником»;

$P(x)$ – « x способствует провозу запрещенных товаров»;

x, y – переменные.

Представим выражения в виде логики предикатов.

Первая посылка: $\forall x((E(x) \& \overline{V(x)}) \rightarrow \exists y(S(x, y) \& C(y)))$.

Вторая посылка: $\exists x(P(x) \& E(x) \& \forall y(S(x, y) \rightarrow P(y)))$.

Третья посылка: $\forall x(P(x) \rightarrow \overline{V(x)})$.

Заключение: $\overline{\exists x(C(x) \& P(x))}$.

Приводим к предваренной форме первую посылку:

$$\begin{aligned} & \forall x \exists y ((E(x) \& \overline{V(x)}) \rightarrow S(x, y) \& C(y)) = \\ & (E(x) \& \overline{V(x)}) \rightarrow S(x, f(x)) \& C(f(x)) = \\ & \overline{(E(x) \& \overline{V(x)}) \vee (S(x, f(x)) \& C(f(x)))} = \\ & (\overline{E(x)} \vee V(x)) \vee (S(x, f(x)) \& C(f(x))) = (\overline{E(x)} \vee V(x)) \vee S(x, f(x)) = \\ & (\overline{E(x)} \vee V(x)) \vee C(f(x)). \end{aligned}$$

Приводим к предваренной форме вторую посылку:

$$P(a) \& E(a) \& (\overline{S(a, y)} \vee P(y)).$$

Приводим к предваренной форме третью посылку: $\overline{P(x)} \vee \overline{V(x)}$.

Заключение: $\overline{P(x)} \vee \overline{C(x)}$.

Представляем посылки и заключение в виде массива родительских предложений:

$$C_1 = \overline{E(x)} \vee V(x) \vee S(x, f(x)).$$

$$C_2 = \overline{E(x)} \vee V(x) \vee C(f(x)).$$

$$C_3 = P(a).$$

$$C_4 = \overline{E(a)}.$$

$$C_5 = \overline{S(a, y)} \vee P(y).$$

$$C_6 = \overline{P(x)} \vee \overline{V(x)}.$$

$$C_7 = \overline{P(x)} \vee \overline{C(x)}.$$

Доказательство противоречивости осуществляем, порождая новые родительские предложения.

$C_8 = \overline{V(a)}$ в результате резолюции C_3 и C_6 по P и подстановки $S(a, x)$.

$C_9 = V(a) \vee C(f(a))$ – резолюция C_2 и C_4 по E и подстановки $S(a, x)$.

$C_{10} = \overline{C(a)}$ – резолюция C_3 и C_7 по P и подстановки $S(a, x)$.

$C_{11} = V(a)$ – резолюция C_9 и C_{10} по C .

Резолюция C_8 и C_{11} по V дает пустую резольвенту. Следовательно, можно сделать вывод, что некоторые из таможенных лиц способствовали провозу запрещенных товаров.

Практические задания

1. Среди следующих формул указать те, которые являются аксиомами исчисления высказываний:

(а) $(F \rightarrow ((F \rightarrow F) \rightarrow F)) \rightarrow ((F \rightarrow (F \rightarrow F)) \rightarrow (F \rightarrow F));$

(б) $F \rightarrow ((\overline{F} \rightarrow G) \rightarrow F);$

(в) $(G \rightarrow H) \rightarrow (F \rightarrow (G \rightarrow H));$

(г) $(\overline{F} \rightarrow G) \rightarrow (\overline{\overline{G}} \rightarrow \overline{\overline{F}});$

(д) $(\overline{F} \rightarrow \overline{\overline{G}}) \rightarrow ((\overline{\overline{F}} \rightarrow \overline{\overline{G}}) \rightarrow F);$

(е) $F \rightarrow (\overline{\overline{F}} \rightarrow F);$

(ж) $(G \rightarrow F) \rightarrow ((G \rightarrow \overline{\overline{F}}) \rightarrow \overline{\overline{G}});$

(з) $(F \rightarrow G) \rightarrow (\overline{\overline{H}} \rightarrow (F \rightarrow G)).$

2. Выяснить, является ли данная последовательность формул выводом из аксиом. Если является, то обосновать каждый шаг построения этой последовательности. Если не является, то доказать это.

- (a) (1) $G \rightarrow (F \rightarrow G)$,
 (2) $(G \rightarrow (F \rightarrow G)) \rightarrow (G \rightarrow (G \rightarrow (F \rightarrow G)))$,
 (3) $(G \rightarrow (G \rightarrow (F \rightarrow G)))$.
- (б) (1) $(\bar{G} \rightarrow \bar{F}) \rightarrow (\bar{G} \rightarrow F) \rightarrow G$,
 (2) $(\bar{F} \rightarrow (\bar{G} \rightarrow \bar{F}))$,
 (3) $(\bar{F} \rightarrow ((\bar{G} \rightarrow F) \rightarrow G))$.
- (в) (1) $F \rightarrow (G \rightarrow F)$,
 (2) $(F \rightarrow (G \rightarrow F)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow F))$,
 (3) $(F \rightarrow G) \rightarrow (F \rightarrow F)$.
- (г) (1) $(F \rightarrow F) \rightarrow (G \rightarrow (F \rightarrow F))$,
 (2) $F \rightarrow F$,
 (3) $G \rightarrow (F \rightarrow F)$.
- (д) (1) $(H \rightarrow F) \rightarrow (G \rightarrow (H \rightarrow F))$,
 (2) $((H \rightarrow F) \rightarrow (G \rightarrow (H \rightarrow F))) \rightarrow$
 $(F \rightarrow ((H \rightarrow F) \rightarrow (G \rightarrow (H \rightarrow F))))$,
 (3) $F \rightarrow ((H \rightarrow F) \rightarrow (G \rightarrow (H \rightarrow F)))$,
 (4) $F \rightarrow ((H \rightarrow F) \rightarrow (G \rightarrow (H \rightarrow F))) \rightarrow$
 $\rightarrow ((F \rightarrow (H \rightarrow F)) \rightarrow (F \rightarrow (G \rightarrow (H \rightarrow F))))$,
 (5) $((F \rightarrow (H \rightarrow F)) \rightarrow (F \rightarrow (G \rightarrow (H \rightarrow F))))$,
 (6) $F \rightarrow (H \rightarrow F)$,
 (7) $F \rightarrow (G \rightarrow (H \rightarrow F))$.
- (е) (1) $\bar{G} \rightarrow \bar{G}$,
 (2) $(\bar{G} \rightarrow \bar{G}) \rightarrow ((\bar{G} \rightarrow G) \rightarrow G)$,
 (3) $((\bar{G} \rightarrow G) \rightarrow G)$.
- (ж) (1) $\bar{\bar{F}} \rightarrow (\bar{\bar{G}} \rightarrow \bar{\bar{F}})$,

3. Доказать, что следующие формулы являются теоремами формализованного исчисления высказываний, построив последовательности формул, являющиеся выводами данных формул из аксиом.

- (a) $F \rightarrow F$;
 (б) $G \rightarrow (F \rightarrow F)$;
 (в) $(F \rightarrow G) \rightarrow (F \rightarrow F)$;
 (г) $G \rightarrow (G \rightarrow (F \rightarrow G))$;
 (д) $G \rightarrow (F \rightarrow (G \rightarrow F))$;
 (е) $G \rightarrow (F \rightarrow (H \rightarrow F))$;

- (ж) $F \rightarrow (G \rightarrow (H \rightarrow F))$;
 (з) $F \rightarrow ((G \rightarrow G) \rightarrow (H \rightarrow F))$;
 (и) $G \rightarrow (F \vee G)$;
 (к) $(\overline{G} \rightarrow G) \rightarrow G$;
 (л) $\overline{G} \rightarrow ((\overline{F} \rightarrow G) \rightarrow F)$.

4. Выяснить, является ли данная последовательность выводом из гипотез. Если да, то указать, выводом из каких формул и какой формулы является. Если нет, то объяснить почему.

- (а) (1) $G \rightarrow H$,
 (2) G ,
 (3) H ,
 (4) $H \rightarrow (F \rightarrow H)$,
 (5) $F \rightarrow H$.
 (б) (1) $F \rightarrow G$,
 (2) $F \rightarrow \overline{G}$,
 (3) $(F \rightarrow G) \rightarrow (F \rightarrow \overline{G}) \rightarrow \overline{F}$,
 (4) $(F \rightarrow \overline{G}) \rightarrow \overline{F}$,
 (5) \overline{F} .
 (в) (1) G ,
 (2) $G \rightarrow (F \rightarrow G)$,
 (3) $F \rightarrow G$.
 (г) (1) $(F \rightarrow (F \rightarrow G)) \rightarrow ((F \rightarrow F) \rightarrow (F \rightarrow G))$,
 (2) $F \rightarrow (F \rightarrow G)$,
 (3) $(F \rightarrow F) \rightarrow (F \rightarrow G)$,
 (4) $F \rightarrow F$,
 (5) $F \rightarrow G$.
 (д) (1) $F \rightarrow G$,
 (2) $\overline{F} \rightarrow G$,
 (3) $(F \rightarrow G) \rightarrow (\overline{G} \rightarrow \overline{F})$,
 (4) $\overline{G} \rightarrow \overline{F}$,
 (5) $(\overline{F} \rightarrow G) \rightarrow (\overline{G} \rightarrow \overline{F})$,
 (6) $\overline{G} \rightarrow \overline{F}$,
 (7) $(\overline{G} \rightarrow \overline{F}) \rightarrow ((\overline{G} \rightarrow \overline{F}) \rightarrow G)$,
 (8) $(\overline{G} \rightarrow \overline{F}) \rightarrow G$,
 (9) G .

- (e) (1) $F \rightarrow (G \rightarrow H)$,
 (2) $(F \rightarrow (G \rightarrow H)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H))$,
 (3) $(F \rightarrow G) \rightarrow (F \rightarrow H)$,
 (4) $G \rightarrow (F \rightarrow G)$,
 (5) G ,
 (6) $F \rightarrow G$,
 (7) $F \rightarrow H$.
- (ж) (1) $F \rightarrow (G \rightarrow F)$,
 (2) $(F \rightarrow (G \rightarrow F)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow F))$,
 (3) $(F \rightarrow G) \rightarrow (F \rightarrow F)$,
 (4) $F \rightarrow G$,
 (5) $F \rightarrow F$.
- (з) (1) $\bar{G} \rightarrow \bar{F}$,
 (2) $(\bar{G} \rightarrow \bar{F}) \rightarrow ((\bar{G} \rightarrow F) \rightarrow G)$,
 (3) $(\bar{G} \rightarrow F) \rightarrow G$,
 (4) $F \rightarrow (\bar{G} \rightarrow F)$,
 (5) $F \rightarrow G$.

5. Доказать, что имеют место следующие выводимости, построив соответствующие выводы из гипотез:

- (a) $G \vdash H \rightarrow (F \rightarrow G)$;
 (б) $F \rightarrow G, (F \rightarrow (G \rightarrow H)), F \vdash H$;
 (в) $F \rightarrow (F \rightarrow G) \vdash F \rightarrow G$;
 (г) $\bar{G} \rightarrow \bar{F}, F \vdash G$;
 (д) $F \vdash H \rightarrow (\bar{G} \rightarrow F)$;
 (e) $\bar{F}, \bar{G} \vdash \bar{F} \rightarrow G$.

6. Установить правильность рассуждений, построив вывод исчисления высказываний:

(a) *Если философ дуалист, то он не материалист. Если он не материалист, то он метафизик. Этот философ дуалист. Следовательно, он метафизик.*

(б) *Если идет дождь, то крыши мокрые. Крыши не мокрые. Следовательно, дождя нет.*

(в) *Если треугольник равносторонний, то его углы равны. Треугольник равносторонний. Следовательно, его углы равны.*

(г) *Этот человек инженер или рабочий. Он не инженер. Следовательно, он рабочий.*

(д) *Если студент занимается не систематически, то он не имеет прочных знаний. Если он не имеет прочных знаний, то он не будет хорошим специалистом. Следовательно, если студент занимается не систематически, то он не будет хорошим специалистом.*

(е) *Это вещество может быть кислотой либо щелочью. Это вещество не щелочь. Следовательно, это кислота.*

(ж) *Если лес дешевый или транспорт в дефиците, то либо растут рубки, либо уменьшаются посадки. Если растут рубки и уменьшаются посадки, то транспорт в дефиците. Известно, что лес дешев и транспорт не в дефиците. Следовательно, рубки растут тогда и только тогда, когда посадки не уменьшаются.*

(з) *Если не было дождей или были заморозки, то урожай плохой. Известно, что урожай хороший, а заморозков не было. Значит, дожди были.*

7. Перечислить свободные и связанные вхождения каждой из переменных в следующих формулах:

(а) $\forall x(P(x)) \rightarrow P(y)$;

(б) $P(x) \rightarrow \exists xQ(x)$;

(в) $\exists x(A(x)) \wedge B(x)$;

(г) $\forall x(P(x) \rightarrow Q(y)) \wedge \exists y(R(x, y, z))$;

(д) $\forall x(\forall z(P(x, z) \rightarrow Q(y)) \wedge \exists y(R(x, y, z)))$;

(е) $\exists x\forall x(Q(x, y, z) \wedge P(y)) \rightarrow R(y, z)$.

8. Даны предикаты $A(x)$ и $B(x)$. Записать словами предложенные формулы C и D :

(а) $A(x) = \langle\langle x - \text{алгоритм} \rangle\rangle$; $B(x) = \langle\langle x \text{ сходится} \rangle\rangle$.

Записать словами: $C = \forall x(A(x) \rightarrow \overline{B(x)})$; $D = \exists x(B(x) \wedge A(x))$.

(б) $A(x) = \langle\langle x - \text{издательство} \rangle\rangle$; $B(x) = \langle\langle x \text{ выпускает учебники} \rangle\rangle$.

Записать словами: $C = \overline{\forall x(A(x) \rightarrow B(x))}$; $D = \exists x(A(x) \wedge \overline{B(x)})$.

(в) $A(x) = \langle\langle x - \text{целое число} \rangle\rangle$; $B(x) = \langle\langle x - \text{рациональное число} \rangle\rangle$.

Записать словами: $C = \overline{\forall x(B(x) \rightarrow A(x))}$; $D = \exists x(A(x) \wedge \overline{B(x)})$.

(г) $A(x) = \langle\langle x - \text{осёл} \rangle\rangle$; $B(x) = \langle\langle x \text{ упрям} \rangle\rangle$.

Записать словами: $C = \exists x(B(x) \wedge A(x))$; $D = \forall x(A(x) \rightarrow B(x))$.

(д) $A(x) = \langle\langle x - \text{дерево} \rangle\rangle$; $B(x) = \langle\langle x \text{ лиственное} \rangle\rangle$.

Записать словами: $C = \overline{\forall x(A(x) \rightarrow B(x))}$; $D = \exists x(A(x) \wedge \overline{B(x)})$.

9. Записать данные суждения в виде формул логики предикатов. Построить отрицание суждений в виде формулы, не содержащей внешних знаков отрицания. Перевести на естественный язык.

(а) *Некоторые моряки прошли все океаны.*

(б) *Каждый студент выполнил хотя бы одну лабораторную работу.*

(в) *Ни одно четное число, большее 2, не является простым.*

(г) *Произведение любых двух простых чисел не является простым числом.*

(д) *Всякое положительное число больше всякого отрицательного числа.*

(е) *Каждый, купивший билет, получит премию.*

(ж) *Всякий равносторонний треугольник является равнобедренным.*

10. Установить правильность рассуждения, построив вывод исчисления предикатов.

(а) *Каждый студент честен. Джон нечестен. Значит, он не студент.*

(б) *Всякий, кто не может решить эту задачу, – не математик. Иван не может решить эту задачу. Значит, Иван не математик.*

(в) *Все металлы теплопроводны. Дерево не теплопроводно. Значит, дерево не металл.*

(г) *Все натуральные числа – целые. 5 – натуральное число. Значит, 5 – целое число.*

(д) *Все дельфины – киты. Ни один кит не является рыбой. Следовательно, все дельфины – не рыбы.*

11. Определить, какие из следующих формул выполнимы, а какие тождественно ложны:

(а) $\forall x(P(x))$;

(б) $\exists x\forall y(Q(x, y) \wedge \overline{Q(x, y)})$;

(в) $\exists x\forall y(Q(x, y) \rightarrow \forall z(R(x, y, z)))$;

(г) $\exists x(P(x)) \rightarrow \forall x(P(x))$;

(д) $\forall x(P(x) \wedge \overline{P(x)})$.

12. Докажите, что следующие формулы являются тавтологиями логики предикатов:

(а) $\exists x(P(x) \vee Q(x)) \leftrightarrow (\exists x(P(x)) \vee \exists x(Q(x)))$;

(б) $\exists x(P(x) \wedge Q(x)) \rightarrow (\exists x(P(x)) \wedge \exists x(Q(x)))$;

- (в) $(\forall x(P(x)) \vee \forall x(Q(x))) \rightarrow \forall x(P(x) \vee Q(x))$;
 (г) $\forall x(P(x) \vee Q(x)) \leftrightarrow \forall x(P(x) \vee Q(x))$;
 (д) $\forall x(P(x) \wedge Q(x)) \leftrightarrow \forall x(P(x) \wedge Q(x))$;
 (е) $\overline{\forall x(P(x))} \leftrightarrow \exists x(\overline{P(x)})$.

13. Найти предваренную нормальную форму формул:

- (а) $\forall x(A(x) \rightarrow \overline{B(y)}) \rightarrow \exists y(B(y) \rightarrow \overline{A(x)})$;
 (б) $\forall x(A(x) \rightarrow \exists x(B(y))) \rightarrow \exists x(\overline{A(x)} \rightarrow \overline{B(y)})$;
 (в) $\forall x(\overline{A(x)} \rightarrow \exists x(\overline{C(y)})) \rightarrow \forall x(C(x) \rightarrow A(x))$;
 (г) $\forall x(\overline{A(x)} \rightarrow \exists x(B(y))) \rightarrow (\overline{B(y)} \rightarrow A(x))$;
 (д) $\forall x(\overline{A(x)} \rightarrow \forall x(\overline{B(x)})) \rightarrow (\overline{B(x)} \vee A(x))$;
 (е) $\exists x(A(x) \rightarrow \forall y(B(y)))(\overline{A(x)} \rightarrow \forall y(B(y))) \rightarrow B(y)$.

14. Проверить вывод методом резолюций:

- (а) $A \rightarrow (B \vee C), A, B \rightarrow D, C \rightarrow D \vdash D$;
 (б) $\overline{A} \rightarrow (B \vee C), \overline{A} \vee C, \overline{B} \vdash C$;
 (в) $A \wedge C \rightarrow B, A, B \rightarrow D, C \vdash D$;
 (г) $A \vee B, A \rightarrow C, B \rightarrow D \vdash C \vee D$;
 (д) $A, C, A \wedge C \rightarrow D, D \rightarrow B \vdash B$;
 (е) $A \rightarrow B, C \rightarrow A, \overline{C} \rightarrow D, \overline{D} \vdash B$.

15. Доказать противоречивость формулы логики предикатов:

- (а) $(P(x) \vee \overline{S(x)} \vee \overline{H(x)})(\overline{R(y)} \vee \overline{S(y)})R(z)\overline{P(z)}(\overline{H(v)} \vee \overline{E(v)})\overline{E(w)}$;
 (б) $(\overline{P(x)} \vee \overline{V(x)})(\overline{P(x)} \vee \overline{C(x)})(\overline{E(x)} \vee \overline{V(x)} \vee \overline{S(y,x)})\overline{E(a)}(\overline{S(y,a)} \vee \overline{P(y)})$;
 $(\overline{E(x)} \vee \overline{V(x)} \vee \overline{C(y)})\overline{P(a)}\overline{E(a)}(\overline{P(x)} \vee \overline{V(x)})(\overline{P(x)} \vee \overline{C(x)})$;
 (в) $\overline{P(a)}(\overline{E(x)} \vee \overline{V(x)} \vee \overline{S(y,x)})\overline{E(a)}(\overline{S(y,a)} \vee \overline{P(y)})(\overline{P(x)} \vee \overline{V(x)})$;
 $(\overline{C(x)} \vee \overline{D(x)})(\overline{A(x)} \vee \overline{C(y)})(\overline{A(x)} \vee \overline{D(x)} \vee \overline{B(x)})(\overline{B(x)} \vee \overline{C(y)})\overline{D(y)}$.

16. Используя метод резолюций в логике предикатов, доказать логичность следующих рассуждений:

(а) *Все студенты нашей группы – члены клуба «Спартак». А каждый член клуба «Спартак» занимается спортом. Следовательно, все студенты нашей группы занимаются спортом.*

(б) *Некоторые из первокурсников знакомы со всеми второкурсниками, а некоторые из второкурсников – спортсмены. Следовательно, ряд первокурсников знаком с некоторыми спортсменами.*

(в) *Некоторые пациенты любят своих докторов. Ни один пациент не любит знахаря. Следовательно, никакой доктор не является знахарем.*

(г) *Ни один человек не является четвероногим. Все мужчины – люди. Следовательно, ни один мужчина не является четвероногим.*

(д) *Все металлы теплопроводны. Дерево не теплопроводно. Значит, дерево не металл.*

(е) *Все медсестры – медицинские работники. Все медицинские работники имеют право на льготы. Следовательно, все медсестры имеют право на льготы.*

3. ТЕОРИЯ АЛГОРИТМОВ

3.1. Основные понятия теории алгоритмов

Теория алгоритмов – это наука, изучающая общие свойства и закономерности алгоритмов, разнообразные формальные модели их представления. На основе формализации понятия алгоритма возможно сравнение алгоритмов по их эффективности, проверка их эквивалентности, определение областей применимости.

Предметом теории алгоритмов являются объекты и действия над ними в смысле их точного определения, комплексы действий их свойств, а также установление того, что можно и что нельзя получить в результате их действия.

Цель теории алгоритмов – доказательство алгоритмической разрешимости или неразрешимости той или иной математической проблемы.

Алгоритмическая разрешимость – существование однозначного и точного (в соответствии с некоторым критерием) решения. Основной чертой алгоритмического подхода является его *конечность* (*финитность*), заключающаяся в рассмотрении только конечного комплекса действий над конечным числом объектов.

Рассмотрим неформальное определение алгоритма. *Алгоритм* – это заданное на некотором языке конечное предписание, задающее конечную последовательность выполнимых элементарных операций для решения задачи.

Пусть D – область (множество) исходных данных задачи, а R – множество возможных результатов, тогда мы можем говорить, что алгоритм осуществляет отображение $D \rightarrow R$. Поскольку такое отображение может быть неполным, то вводятся следующие понятия.

Алгоритм называется *частичным алгоритмом*, если мы получаем результат только для некоторых $d \in D$, и *полным алгоритмом*, если алгоритм получает правильный результат для всех $d \in D$.

Несмотря на усилия исследователей, отсутствует одно исчерпывающе строгое определение понятия «алгоритм», в теории алгоритмов были введены различные формальные определения алгоритма, и удивительным научным результатом является доказательство эквивалентности этих формальных определений в смысле их равносильности. Первые работы по уточнению понятия «алгоритм» появились

в 1936–1937 гг. Это были работы Тьюринга, Поста, Маркова, Чёрча. Рассмотрим варианты словесного определения алгоритма, принадлежащие российским ученым А. Н. Колмогорову и А. А. Маркову.

Определение Колмогорова. *Алгоритм* – это всякая система вычислений, выполняемых по строго определенным правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи.

Определение Маркова. *Алгоритм* – это точное предписание, определяющее вычислительный процесс, идущий от варьируемых исходных данных к искомому результату.

Рассмотрим *основные требования к алгоритмам*.

1. Алгоритм применяется к исходным данным и дает результаты. Кроме того, в процессе работы алгоритма могут появляться промежуточные данные.

2. Данные для своего размещения требуют памяти. Память состоит из ячеек, так что каждая ячейка может содержать один символ алфавита данных. Таким образом, объем данных и требуемая память согласованы.

3. Алгоритм состоит из отдельных элементарных шагов или действий. Множество различных шагов, из которых составлен алгоритм, конечно.

4. Последовательность шагов алгоритма детерминирована, т. е. после каждого шага либо указывается, какой шаг делать дальше, либо дается команда остановки, после чего работа алгоритма считается законченной.

5. Результативность алгоритма – остановка после конечного числа шагов с указанием того, что считать результатом.

Любая практическая задача требует предварительного задания исходных данных. Как правило, можно задать некоторое характерное число n . Например, для задачи сортировки массива чисел по возрастанию n – число чисел в массиве, для задачи решения системы линейных уравнений n – число уравнений. Характерное число задачи определяет размерность задачи как величину массива исходных данных.

С ростом характерного числа размерность задачи возрастает. Рассмотрим понятие скорости роста для функций, зависящих от целочисленного параметра n .

Функции $f(n)$ и $g(n)$ имеют *одинаковую скорость роста*, если при достаточно больших n , начиная с некоторого n_0 , выполняется условие

$$C_1g(n) \leq f(n) \leq C_2g(n),$$

где C_1, C_2 – некоторые константы.

Скорость роста функции $f(n)$ *ограничена снизу* скоростью роста функции $g(n)$, если при достаточно больших n , начиная с некоторого n_0 , выполняется условие $C_1g(n) \leq f(n)$, где C_1 – некоторая константа.

Скорость роста функции $f(n)$ *ограничена сверху* скоростью роста функции $g(n)$, если при достаточно больших n , начиная с некоторого n_0 , выполняется условие $f(n) \leq C_2g(n)$, где C_2 – некоторая константа.

Скорость роста функции $f(n)$ *больше* скорости роста функции $g(n)$, если для любой сколь угодно большой константы C_2 существует некоторое n_0 , начиная с которого выполняется условие $f(n) \geq C_2g(n)$.

Для того чтобы более наглядно представить скорости роста функций, их сравнивают со скоростями роста хорошо известных функций. В качестве таких функций чаще всего используют степенные функции n^a . При $a = 1$ скорость роста функции n^a называют *линейной*, при $a = 2$ – квадратичной, при $a = 3$ – кубической и т. д. Скорость роста вида n^a называют *полиномиальной*. Очевидно, что при возрастании a скорость роста тоже увеличивается. Для некоторых функций скорости роста превосходят в пределе при $n \rightarrow \infty$ любую полиномиальную скорость. Такими функциями являются, например, 2^n , e^n , $n!$. Скорости такого типа называют *экспоненциальными*.

Обозначим через $r(n)$ скорость роста размерности задачи. В задаче вычисления таблицы значений булевой функции n переменных скорость роста определяется таблицей значений переменных. Так как различных наборов переменных 2^n , а каждый набор состоит из n символов, то размерность задачи равна $n2^n$ и скорость роста будет экспоненциальной. В задаче решения системы линейных алгебраических уравнений методом исключения Гаусса наиболее быстро растет число элементов матрицы системы уравнений размером $n \times n$, поэтому скорость роста размерности этой задачи будет квадратичной, $r(n) = n^2$.

Размерность задачи определяет память, необходимую для представления исходных данных в ЭВМ. Кроме того, необходима дополнительная память для размещения промежуточных данных. Величина этой памяти зависит от конкретного алгоритма, ее, как правило, трудно рассчитать.

Рассмотрим время реализации алгоритма – время счета. Пусть при выполнении некоторого алгоритма выполняются элементарные

операции t_1, t_2, \dots, t_k (арифметические, логические и другие). Среднее время выполнения этих операций обозначим через t_1, t_2, \dots, t_k . По аналогии с размерностью задачи введем понятие скорости роста числа выполняемых операций в зависимости от характерного числа n . Обозначим их для операций t_1, t_2, \dots, t_k через $g_1(n), g_2(n), \dots, g_k(n)$. Без доказательства приведем следующее утверждение.

При $n \rightarrow \infty$ скорость роста общего времени счета $T(n)$ равна максимальной из скоростей роста числа элементарных операций $g_1(n), g_2(n), \dots, g_k(n)$ независимо от среднего времени их выполнения t_1, t_2, \dots, t_k .

Скорость роста общего времени счета $T(n)$ называется *вычислительной сложностью*, или просто *сложностью алгоритма*.

Обозначим сложность алгоритма через $f(n)$. В зависимости от сложности все алгоритмы делятся на несколько классов.

Полиномиальными называются алгоритмы, сложность которых ограничена некоторым полиномом.

Например, рассмотрим задачу определения максимального элемента в массиве из n чисел. Поскольку число операций сравнения постоянно и равно $n - 1$, сложность алгоритма $f(n) = n$.

Экспоненциальными называются алгоритмы, сложность которых при возрастании n превышает полином любой степени.

Например, рассмотрим задачу коммивояжера. Необходимо обойти n городов и вернуться в исходный пункт, так чтобы суммарный путь был минимальным. Количество всех возможных вариантов обхода равно $0,5n!$ Следовательно, сложность точного решения, основанного на переборе всех вариантов, равна $f(n) = n!$ Данный алгоритм имеет экспоненциальную точность.

Экспоненциальные алгоритмы практически могут быть реализованы только при малых значениях n (обычно при $n < 10$).

Задачи, для которых существуют точные алгоритмы решения полиномиальной сложности, называются *задачами класса P*.

Задачи, для которых не удастся найти точные алгоритмы решения полиномиальной сложности, составляют *класс NP*.

Для многих задач класса *NP* выполняется свойство сводимости, состоящее в том, что данный алгоритм выражается при помощи полиномиального числа операций через другой алгоритм, имеющий полиномиальную сложность.

3.2. Машина Тьюринга

Машина Тьюринга – это наиболее ранняя алгоритмическая модель, созданная в 1937 г. Аланом Тьюрингом, за 9 лет до появления первой ЭВМ.

Машина Тьюринга является расширением модели конечного автомата. Это математическая модель идеализированного вычислительного устройства.

Машина Тьюринга полностью определяется:

- *внешним алфавитом* $A = \{a_0, a_1, \dots, a_n, *\}$, где a_0 – символ пустой ячейки; $*$ – разделитель; $a_1 = 1$;
- *алфавитом внутренних состояний* $Q = \{q_0, q_1, \dots, q_m\}$, где q_0 – состояние остановки машины Тьюринга; q_1 – начальное состояние;
- *программой (функциональной схемой)*, т. е. совокупностью выражений $T(i, j)$ ($i = 1, \dots, m; j = 0, 1, \dots, n$), каждое из которых имеет один из следующих видов: $q_i a_j \rightarrow q_k a_l R$, $q_i a_j \rightarrow q_k a_l L$, где $0 \leq k \leq m, 0 \leq l \leq n$. Выражения $T(i, j)$ называют *командами*.

Устройство и работу машины Тьюринга наглядно можно представить следующим образом: машина Тьюринга состоит из ленты и управляющего устройства со считывающей и записывающей головкой (кареткой) (рис. 7).

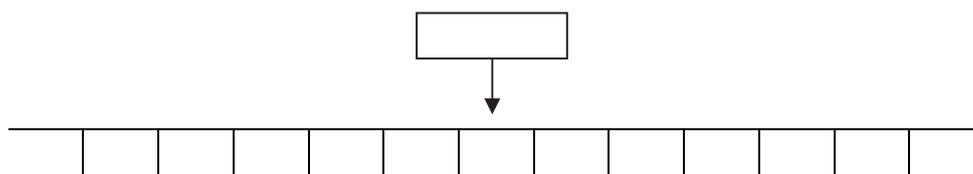


Рис. 7. Машина Тьюринга

Машина имеет бесконечную в обе стороны ленту. Иногда считают, что лента жестко закреплена слева и бесконечна справа. Лента разделена на ячейки, которые нумеруются натуральными числами $1, 2, \dots$. В каждую ячейку ленты заносятся символы внешнего алфавита машины Тьюринга.

В каждый момент времени (такт работы) машина находится в одном из состояний, обозначаемых буквами алфавита внутренних состояний, и «обозревает» одну ячейку ленты (стоит против этой ячейки ленты). По команде $q_i a_j \rightarrow q_k a_l X$, означающей, что машина

находится в состоянии q_i и обзореваает ячейку в которой записана буква a_j , машина переходит в состояние q_k , в обозреваемой ячейке стирает букву a_j и заносит туда a_l . Затем машина переходит к обозрению ячейки, следующей справа или слева от предыдущей, или продолжает обзореваать предыдущую ячейку, в зависимости от того $X=R$ или $X=L$, или место X в команде не заполнено. На следующем такте машина переходит к выполнению команды $q_k a_l \rightarrow q_r a_s X$ и т. д.

Словом в алфавите A или в алфавите Q называется любая последовательность букв соответствующего алфавита. Количество символов в слове называется *длиной слова*.

Под k -й *конфигурацией* понимается изображение ленты машины с информацией, сложившейся на ней к началу k -го такта (или слово в алфавите A , записанное на ленте к началу k -го такта), с указанием обозреваемой ячейки и состояния машины.

Непустое слово w в алфавите A воспринимается машиной в *стандартном положении*, если слово записано в последовательных ячейках ленты, все другие ячейки пусты и машина обзореваает крайнюю справа ячейку в слове w . Стандартное положение называется *начальным (заключительным)*, если машина, воспринимающая слово в стандартном положении, находится в начальном состоянии q_1 (в состоянии остановки q_0).

Например, пусть на ленте записано слово « $abcde$ », управляющее устройство находится в состоянии q_i , и каретка стоит против символа d . Конфигурация в этом случае запишется следующим образом « $abcq_i de$ ».

Так как машина Тьюринга имеет конечный алфавит и конечное число внутренних состояний, то очевидно, что она может выполнять конечное число действий.

Например, машина с внешним алфавитом $A = \{1, a\}$, алфавитом внутренних состояний $Q = \{q_1, q_2\}$ и программой $q_1 1 \rightarrow q_1 1R$, $q_1 a \rightarrow q_1 1R$ из любой начальной конфигурации будет работать бесконечно, заполняя единицами всю ленту вправо от начальной точки.

Порядок работы машины Тьюринга часто задается в виде таблицы. В каждый столбец верхней строчки заносятся символы внутреннего алфавита, в каждую строчку первого столбца – символы внешнего алфавита. В ячейках на пересечении других столбцов и строчек помещаются команды.

Если на пересечении какой-либо строки и какого-либо столбца мы получим пустую клетку, то это означает, что в данном внутреннем состоянии данный символ встретиться не может. Пример представления работы машины Тьюринга представлен в табл. 27.

Таблица 27

A	Q				
	q_1	...	q_i	...	q_m
a_0					
...					
a_j			$q_i a_j X$		
...					
a_n					

Работа машины на основании заданной программы происходит следующим образом. Предположим, что в данный момент времени машина Тьюринга находится во внутреннем состоянии q_i , а в обозреваемой кареткой ячейке ленты находится символ a_j . Тогда машина переходит к выполнению команды $q_i a_j X$ в ячейке, на пересечении столбца q_i и строки a_j :

- в текущую ячейку ленты заносится новый символ a_j (возможно, тот же самый);
- происходит сдвиг головки, или головка остается на месте, т. е. происходит остановка машины;
- машина переходит в новое внутреннее состояние q_i .
- Возможные случаи остановки машины Тьюринга:
 - в ходе выполнения программы машина дойдет до выполнения команды остановки, программа в этом случае считается выполненной, машина останавливается – происходит результативная остановка;
 - машина никогда не останавливается, происходит зацикливание.

Пример

Пусть внешний алфавит $A = \{0, 1, 2\}$, а множество внутренних состояний состоит лишь из одного состояния $Q = \{q_1\}$. Необходимо построить машину Тьюринга, которая в произвольной записи, начиная из любой ячейки, двигаясь вправо, находит первый ноль и останавливается.

Такая машина может быть задана табл. 28.

Таблица 28

A	Q
	q_1
0	q_10
1	q_11R
2	q_12R

Действительно, пусть, например, вначале машина находится в состоянии, представленном на рис. 8.

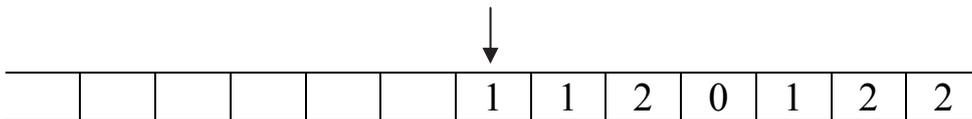


Рис. 8. Начальное состояние машины

Головка обозревает символ 1. В соответствии с табл. 28 выполняется команда q_11R , т. е. в обозреваемую ячейку записывается тот же самый символ 1, а головка смещается вправо (рис. 9).

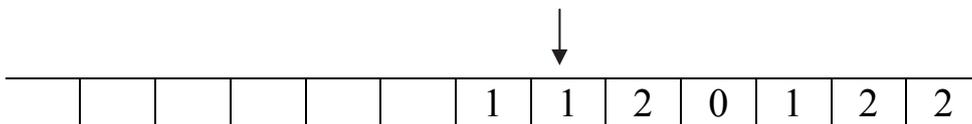


Рис. 9. Состояние на первом такте работы машины

Теперь головка снова обозревает символ 1, в соответствии с табл. 28 выполняется команда q_11R , т. е. в обозреваемую ячейку записывается тот же самый символ 1, а головка смещается вправо (рис. 10).

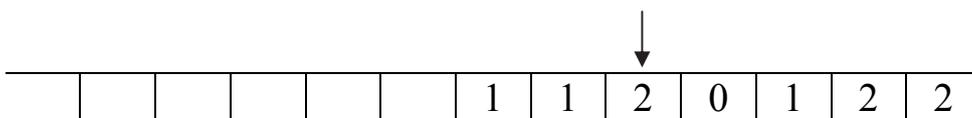


Рис. 10. Состояние на втором такте работы машины

Теперь головка обозревает символ 2, в соответствии с табл. 28 выполняется команда q_12R , т. е. в обозреваемую ячейку записывается тот же самый символ 2, а головка смещается вправо (рис. 11).

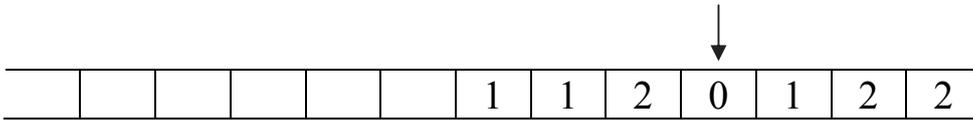


Рис. 11. Состояние на третьем такте работы машины

Теперь головка обозревает символ 0, и в соответствии с табл. 28 выполняется команда q_10 , т. е. в обозреваемую ячейку записывается тот же самый символ 0, а машина останавливается.

Пример

Построим машину Тьюринга, которая производит сложение двух натуральных чисел a и b . Натуральное число a представляется словом $1\dots1 = 1^a$, состоящим из a единиц (числу 3 соответствует слово 111). Сложить два числа a и b – это значит слово $1^a a_0 1^b$ преобразовать в слово 1^{a+b} . Это можно сделать, удалив в записи $a*b$ символ разделителя «*» и сдвинув первое слагаемое ко второму. Такая машина может быть задана табл. 29. Внешний алфавит $A = \{1, a_0, *\}$. Множество внутренних состояний состоит из трех состояний $Q = \{q_0, q_1, q_2\}$.

Таблица 29

A	Q	
	q_1	q_2
a_0	–	–
1	$q_2 a_0 L$	$q_2 1 L$
*	$q_0 a_0$	$q_0 1$

Последовательность конфигураций для случая $a = 1, b = 2$:

$$1*1q_11 \Rightarrow 1*q_21a_0 \Rightarrow 1q_2*1a_0 \Rightarrow 1q_011a_0.$$

3.3. Вычислимые по Тьюрингу функции

Будем рассматривать функции f от одной или нескольких переменных, заданных на множестве $N = \{0, 1, 2, \dots, n, \dots\}$ натуральных

чисел или его подмножествах (частичные функции) и принимающие значения на множестве N .

Функция $f(x_1, x_2, \dots, x_n)$ называется *вычислимой*, если существует алгоритм, позволяющий вычислять ее значения для тех переменных, для которых она определена, и работающий бесконечно, если функция для данного набора переменных не определена.

Функция $f(x_1, x_2, \dots, x_n)$ называется *вычислимой по Тьюрингу*, если существует машина Тьюринга, вычисляющая эту функцию.

Функция $f(x_1, x_2, \dots, x_n)$ называется *правильно вычислимой по Тьюрингу*, если начальное слово $01^{x_1}01^{x_2}0\dots01^{x_n}0$ она переводит в слово $q_001^{f(x_1, x_2, \dots, x_n)}0\dots0$ и при этом в процессе работы не пристраивает к начальному слову новых ячеек на ленте ни слева, ни справа. Если же функция f не определена на данном наборе значений аргументов, то, начав работать из указанного положения, она никогда в процессе работы не будет надстраивать ленту слева.

Пусть заданы машины Тьюринга O_1 и O_2 , имеющие общий внешний алфавит $\{a_0, a_1, \dots, a_m\}$ и алфавиты внутренних состояний $\{q_0, q_1, \dots, q_n\}$ и $\{q_0, q_1, q_2, \dots, q_t\}$ соответственно.

Композицией (или *произведением*) машины O_1 на машину O_2 называется новая машина O с тем же внешним алфавитом, внутренним алфавитом $\{q_0, q_1, \dots, q_n, q_{n+1}, q_{n+2}, \dots, q_{n+t}\}$ и программой, получающейся следующим образом. Во всех командах из O_1 , содержащих символ остановки q_0 , заменяем его на q_{n+1} . Все остальные символы в командах из O_1 остаются неизменными. В командах из O_2 символ q_0 оставляем неизменным, а все остальные состояния q_i ($i = 1, \dots, t$) заменяем соответственно на q_{n+i} . Совокупность всех так полученных команд образует программу машины-композиции O .

Тезис Тьюринга. Всякий алгоритм можно реализовать машиной Тьюринга.

Тезис Тьюринга доказать нельзя. Это утверждение означает, что математическое понятие вычислимой по Тьюрингу функции является идеальной моделью интуитивного понятия алгоритма. Этот тезис подтверждается опытом. По своему характеру тезис Тьюринга напоминает математические законы механики, которые точно так же не могут быть доказаны, но, открытые Ньютоном, многократно подтверждены опытом. В силу тезиса Тьюринга невозможность построения машины Тьюринга означает отсутствие алгоритма решения данной проблемы.

Изучение машин Тьюринга закладывает фундамент алгоритмического мышления, сущность которого состоит в том, что нужно уметь разделять процесс вычисления на простые составляющие шаги. В машине Тьюринга такое разделение доведено до предельной простоты.

В современной ЭВМ алгоритмический процесс разделяется не на столь мелкие составляющие, как в машине Тьюринга. Наоборот, есть стремление укрупнить выполняемые машиной процедуры. Например, операция сложения в машине Тьюринга – целая программа, а в ЭВМ – это простейшая функция.

Практические задания

1. Определить, в какое слово перерабатывает машина Тьюринга, с программой $q_1 a_0 \rightarrow q_0 1, q_1 1 \rightarrow q_1 1R$ заданное слово, если она находится в начальном состоянии q_1 и обозревает указанную ячейку (обозреваемые ячейки необходимо считать слева). Изобразить схематически последовательность конфигураций, возникающих на ленте на каждом такте работы машины.

(а) $11a_0111a_01$ (ячейка 2);

(б) 11111 (ячейка 5);

(в) $1a_0 a_0111$ (ячейка 3);

(г) a_0111a_0111 (ячейка 1);

(д) $1111a_011$ (ячейка 3);

(е) $1a_0 a_0111$ (ячейка 5);

(ж) $11a_01111$ (ячейка 1);

(з) $1a_01111a_01$ (ячейка 3);

(и) 1111111 (ячейка 4);

(к) $11a_0a_0a_0111$ (ячейка 4).

2. Сконструировать машину Тьюринга, преобразующую заданное слово w_1 в слово w_2 , при условии, что машина начинает работать из начального стандартного состояния. Изобразить схематически последовательность конфигураций, возникающих на ленте на каждом такте работы машины.

(а) $w_1 = 000, w_2 = 0000$;

(б) $w_1 = 010, w_2 = 0101$;

(в) $w_1 = 000, w_2 = 0001$;

(г) $w_1 = 011, w_2 = 1010$;

(д) $w_1 = 001, w_2 = 111$;

(е) $w_1 = 011, w_2 = 1011$;

- (ж) $w_1 = 001$, $w_2 = 110$;
 (з) $w_1 = 011$, $w_2 = 010$;
 (и) $w_1 = 010$, $w_2 = 0100$;
 (к) $w_1 = 110$, $w_2 = 1100$.

3. Имеется машина Тьюринга с внешним алфавитом $A = \{a_0, 1\}$, алфавитом внутренних состояний $Q = \{q_0, q_1\}$ и функциональной схемой (программой).

A	Q	
	q_0	q_1
a_0		$q_01\Pi$
1	$q_2a_0Л$	$q_11\Pi$

В столбце q_0 ничего не указано, так как q_0 – заключительное состояние машины, т. е. такое состояние, оказавшись в котором, машина останавливается. Функциональную схему, или программу, кратко можно записать в виде последовательности из двух команд: $q_1a_0 \rightarrow q_01$, $q_11 \rightarrow q_11\Pi$.

Необходимо определить, в какое слово перерабатывает машина каждое из следующих слов, если она находится в начальном состоянии q_1 и обозначает указанную ячейку:

- (а) $1a_011a_0a_011$ (обозревается ячейка 4, считая слева);
 (б) $11a_0111a_01$ (обозревается ячейка 2);
 (в) $1a_0a_0111$ (обозревается ячейка 3);
 (г) $1111a_011$ (обозревается ячейка 4);
 (д) $11a_01111$ (обозревается ячейка 3);
 (е) 1111111 (обозревается ячейка 4);
 (ж) 11111 (обозревается ячейка 5);
 (з) $111\dots 1$ (k единиц, обозревается k -я ячейка).

Изобразить схематически последовательность конфигураций, возникающих на ленте на каждом такте работы машины.

4. Имеется машина Тьюринга с внешним алфавитом $A = \{a_0, 1\}$, алфавитом внутренних состояний $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$ и со следующей функциональной схемой (программой):

A	Q						
	q_1	q_2	q_3	q_4	q_5	q_6	q_7
a_0	$q_4a_0\Pi$	$q_6a_0\Pi$	$q_6a_0\Pi$	q_41	$q_4a_0\Pi$	q_0a_0	$q_6a_0\Pi$
1	$q_21Л$	$q_31Л$	$q_11Л$	q_5a_0	q_5a_0	q_7a_0	q_7a_0

Изображая на каждом такте работы машины получившуюся конфигурацию, определить, в какое слово перерабатывает машина каждое из следующих слов, исходя из начального стандартного положения:

- (а) 11111;
- (б) 111111;
- (в) 1111;
- (г) 1111111;
- (д) 111;
- (е) $1a_0111a_0a_01111$;
- (ж) $11a_0a_0111111$;
- (з) $11a_0111$.

5. Машина Тьюринга задается следующей функциональной схемой:

A	Q		
	q_1	q_2	q_3
a_0		$q_31\Pi$	$q_1a_0\Pi$
1	$a_2a_0\Pi$	$q_21\Pi$	$q_31\Pi$
*	q_0a_0	$q_2*\Pi$	$q_3*\Pi$

Определить, в какое слово перерабатывает машина каждое из следующих слов, исходя из начального стандартного состояния. После этого необходимо выявить общую закономерность в работе машины:

- (а) 111*111;
- (б) 1111*11;
- (в) 111*1;
- (г) 1*11;
- (д) 11*111;
- (е) 11111*;
- (ж) *1111.

6. Машина Тьюринга определяется следующей функциональной схемой:

A	Q			
	q_1	q_2	q_3	q_4
a_0	$q_1a_0\Pi$	$q_4a_0\Pi$	$q_3a_0\Pi$	$q_1a_0\Pi$
1	$q_3a_0\Pi$	$q_21\Pi$	$q_4a_0\Pi$	$q_41\Pi$
*	q_0a_0	$q_3*\Pi$		$q_4*\Pi$

Необходимо определить, в какое слово перерабатывает машина каждое из следующих слов, исходя из начального стандартного состояния:

- (а) 111*11;
- (б) 11*11;
- (в) 1111*1;
- (г) 11111*111;
- (д) 11111*1111.

После этого необходимо выявить общую закономерность в работе машины.

7. Машина Тьюринга определяется следующей функциональной схемой:

A	Q			
	q_1	q_2	q_3	q_4
a_0	$q_4 a_0 \Pi$	$q_3 a_0 \Pi$	$q_1 a_0 \Pi$	$q_0 a_0 \Pi$
1	$q_2 \alpha$	$q_1 \beta$	$q_1 \Pi$	$q_1 1 \Pi$
α	$q_1 \alpha \Pi$	$q_2 \alpha \Pi$	$q_3 1 \Pi$	$q_4 a_0 \Pi$
β	$q_1 \beta \Pi$	$q_2 \beta \Pi$	$q_3 a_0 \Pi$	$q_4 1 \Pi$

Для следующих слов необходимо определить, в какое слово переработается каждое из них данной машиной, исходя из изначального положения, при котором машина находится в состоянии q_1 и обозревается указываемая ячейка:

- (а) 11111 (обозревается ячейка 2, считая слева);
- (б) 111 (обозревается ячейка 1);
- (в) 1111111111 (обозревается ячейка 4);
- (г) 111111 (обозревается ячейка 2);
- (д) 1111111111111111 (обозревается ячейка 6).

8. Проверьте, что машина Тьюринга с внешним алфавитом $A = \{a_0, 1\}$ и программой (записанной в виде сокращенной таблицы).

A	Q									
	q_1	q_2	q_3	\bar{q}_1	q_4	\bar{q}_2	q_5	\bar{q}_3	q_6	q_7
a_0	$\bar{q}_1 \Pi$	$\bar{q}_2 \Pi$	$\bar{q}_3 \Pi$	$q_0 a_0$	$\bar{q}_1 \Pi$	$q_0 1$	$\bar{q}_2 \Pi$	$q_7 1$	$\bar{q}_3 \Pi$	$q_0 1$
1	$q_2 \Pi$	$q_3 \Pi$	$q_1 \Pi$	$q_4 a_0$		$q_5 a_0$		$q_6 a_0$		$q_7 \Pi$

каждое слово длиной n в алфавите $A = \{1\}$ перерабатывает в слово длиной r , где r – остаток от деления n на 3. Для понимания действия данного алгоритма, можно применить его к словам 111, 1111, 11111.

9. Машина Тьюринга с внешним алфавитом $A = \{a_0, 1\}$ определяется следующей программой:

A	Q		
	q_1	q_2	q_3
a_0	$q_2 a_0 \Pi$	$q_2 a_0 \Pi$	$q_0 a_0$
1	$q_1 1 \Pi$	$q_3 1 \Pi$	$q_3 1 \Pi$

Остановится ли когда-нибудь эта машина, если она начнет перерабатывать следующее слово (в начальный момент, в состоянии q_1 машина обозревает ячейку, в которой записана самая левая буква перерабатываемого слова)?

- (а) $111a_0a_01$;
- (б) $11a_0a_0111$;
- (в) 111111 ;
- (г) $1a_0a_0a_0a_01$;
- (д) $11a_0a_011$;
- (е) 1 ;
- (ж) $1a_01a_01a_01$;
- (з) 111 ;
- (и) $1a_01a_01$;
- (к) $11a_011$.

Если остановка происходит, то какое слово получается в результате, какая ячейка и в каком (перед остановкой) состоянии обозревается?

10. Остановится ли когда-нибудь машина Тьюринга, заданная следующей программой:

A	Q		
	q_1	q_2	q_3
a_0	$q_1 \Pi$	$q_3 a_0 \text{Л}$	$q_0 a_0$
1	$q_2 1 \Pi$	$q_1 a_0 1 \Pi$	$q_2 1 \text{Л}$

если она начнет перерабатывать следующее слово, начав в состоянии q_1 обозревать ячейку, в которой записана самая левая буква перерабатываемого слова?

- (а) $1111a_01$;
 (б) 11111 ;
 (в) $1a_01a_01$.

Если машина остановится, то какова ее заключительная конфигурация?

11. Останавливается ли машина Тьюринга с внешним алфавитом $A = \{a_0, 1\}$ и функциональной схемой:

A	Q			
	q_1	q_2	q_3	q_4
a_0	$q_2a_0\Pi$	$q_3a_0Л$	$q_11Л$	q_0a_0
1	$q_21\Pi$	$q_4a_0\Pi$	q_01	$q_2a_0\Pi$

при переработке следующих слов (в начальный момент головка машины обозревает ячейку ленты, в которой записана самая левая буква перерабатываемого слова)?

- (а) $111a_01a_01$;
 (б) 1111 ;
 (в) $1a_01a_01a_01$.

Какое слово получается в результате остановки машины и какая ячейка и в каком состоянии обозревается?

12. Сконструируйте машину Тьюринга с внешним алфавитом $A = \{a_0, 1\}$, которая каждое слово в алфавите $A_1 = \{1\}$ перерабатывает в пустое слово, исходя из стандартного начального положения.

13. Сконструируйте машину Тьюринга с внешним алфавитом $A = \{a_0, 1\}$, которая каждое слово длиной n в алфавите $A_1 = \{1\}$ перерабатывает в слово длиной $n + 1$ в том же алфавите A . Использовать алфавит внутренних состояний из двух букв (см. задачу 3).

14. На ленте машины Тьюринга записаны два набора единиц 1. Они разделены звездочкой *. Составить функциональную схему машины так, чтобы она, исходя из стандартного начального положения, выбрала больший из этих наборов, а меньший стерла. Звездочка должна быть сохранена, чтобы было видно, какой из массивов выбран. Рассмотреть примеры работы этой машины применительно к словам:

- (а) $1*11$;
 (б) $11*1$;
 (в) $11*111$;

- (г) 111*11;
- (д) 11*1111;
- (е) 1111*11.

Указание. Машина может работать, например, следующим образом. Заменить крайнюю правую единицу на α и из состояния q_1 перейти в состояние q_2 , в котором она должна, ничего не меняя, прошагать к крайней левой единице. Здесь, перейдя в состояние q_3 , заменить крайнюю левую единицу на букву α . Далее, перейдя в состояние q_4 , прошагать к крайней правой единице, ничего не меняя. Здесь снова заменить единицу на букву α и вернуться к крайней левой единице и т. д. Дальше программа имеет разветвление. Если, начиная двигаться с правого конца, машина в состоянии q_1 , сделав шаг влево, обозревает ячейку с буквой $*$, то это означает, что единицы правого массива закончились. Следовательно, левый массив больше. Тогда машина, перейдя в состояние q_5 , проходит ячейку с буквой $*$, а во всех следующих ячейках слева проставляет единицы. Затем в состоянии q_6 она возвращается к ячейке с $*$, минует ее и следует дальше вправо, стирая содержимое ячеек (там записаны буквы α). Дойдя до первой пустой ячейки, машина останавливается. Если же, начиная двигаться с левого конца, машина в состоянии q_3 , сделав шаг вправо, обозревает ячейку с буквой $*$, то это означает, что закончились единицы левого массива. Следовательно, большим оказывается правый массив. Привлекая новые состояния q_7 и q_8 , строим программу аналогично предыдущему ответвлению.

15. Построить машину Тьюринга, которая бы к натуральному числу в десятичной системе счисления прибавляла единицу.

16. Построить функциональную схему машины Тьюринга, которая бы от натурального числа в десятичной системе счисления отнимала единицу.

17. Дана конечная совокупность единиц, вписанных в ячейки, взятые подряд без пропусков. Построить функциональную схему такой машины Тьюринга, которая записывала бы в десятичной системе счисления число этих единиц, т. е. пересчитывала бы набор единиц (дешифратор).

18. Доказать, что следующие функции вычислимы по Тьюрингу, для чего построить машины Тьюринга, вычисляющие их:

- (а) $f(x, y) = x + y$;
- (б) $sg(x) = \begin{cases} 0, & \text{если } x = 0, \\ 1, & \text{если } x > 0; \end{cases}$

(в) $(x, y) = x - y \ (x \geq y)$;

(г) $f(x) = \frac{x}{2}$;

(д) $f(x) = \left[\frac{x}{2} \right]$ – целая часть числа $\frac{x}{2}$;

(е) $f(x) = 2x + 1$;

(ж) $f(x) = \frac{1}{x-2}$;

(з) $f(x) = 2^{1-x}$.

19. Доказать, что следующие функции вычислимы по Тьюрингу, построив соответствующие машины Тьюринга:

(а) $f_3(x) = \begin{cases} 1, & x \text{ делится на } 3, \\ 0, & x \text{ не делится на } 3; \end{cases}$

(б) $f_p(x) = \begin{cases} 1, & x \text{ делится на } p, \\ 0, & x \text{ не делится на } p. \end{cases}$

20. Машина Тьюринга имеет следующую функциональную схему: $q_10 \rightarrow q_01Л$, $q_11 \rightarrow q_11П$, $q_20 \rightarrow q_31П$, $q_21 \rightarrow q_21Л$, $q_30 \rightarrow q_00$, $q_31 \rightarrow q_01$. Найти формульное выражение функции $f(x)$, вычисляемой этой машиной.

21. По программе машины Тьюринга записать формульное выражение функции $f(x, y)$, вычисляемой этой машиной:

A	Q					
	q_1	q_2	q_3	q_4	q_5	q_6
0	$q_20П$	$q_10Л$	$q_40Л$	$q_40Л$	$q_60Л$	q_00
1	$q_11П$	$q_30П$	$q_30П$	$q_51Л$	$q_51Л$	q_01

22. Доказать, что следующие функции примитивно рекурсивны, руководствуясь непосредственно определением примитивно рекурсивной функции:

(а) $f(x) = n$;

(б) $f(x) = x + n$;

(в) $f(x) = x + y$;

(г) $f(x) = xy$.

23. Доказать, что функция, полученная суперпозицией примитивно рекурсивных функций, сама примитивно рекурсивна.

24. Доказать, что следующие функции примитивно рекурсивны, руководствуясь определением примитивно рекурсивной функции и теоремой, установленной в предыдущей задаче:

(а) $f(x) = x!$ (здесь $0! = 1$);

(б) $f(x, y) = x^y$ (здесь $0^0 = 1$);

(в) $sg(x) = \begin{cases} 0, & \text{если } x = 0, \\ 1, & \text{если } x > 0. \end{cases}$

25. Доказать, что следующие функции примитивно рекурсивны:

(а) $q(x, y) = \left[\frac{x}{y} \right]$ – целая часть дроби $\frac{x}{y}$ (здесь $\left[\frac{x}{y} \right] = 0$);

(б) $q(x, y) = \left[\frac{x}{y} \right]$ – целая часть дроби $\frac{x}{y}$ (здесь $\left[\frac{x}{0} \right] = 0$);

(в) $r(x, y)$ – остаток от деления y на x (здесь $r(x, 0) = x$);

(г) $\tau(x)$ – число делителей числа x , где $\tau(0) = 0$;

(д) $\sigma(x)$ – число делителей числа x , где $\sigma(x) = 0$;

(е) $\text{tr}(x)$ – число делителей числа x , где $\text{tr}(x) = 0$;

(ж) $\pi(x)$ – число простых чисел, не превосходящих x .

26. Доказать, что примитивно рекурсивны булевы функции:

(а) отрицание \bar{x} ;

(б) конъюнкция $x \wedge y$;

(в) дизъюнкция $x \vee y$;

(г) все булевы функции.

ЗАКЛЮЧЕНИЕ

В учебном пособии «Математическая логика и теория алгоритмов» систематизировано изложены темы, традиционно изучаемые в курсе математической логики и теории алгоритмов: алгебра логики и исчисление высказываний, логика и исчисление предикатов, формальные аксиоматические теории, теория алгоритмов и теория вычислительной сложности. В пособии представлены как основные теоретические положения, так и примеры решения практических задач.

Материал данного учебного пособия предназначен для формирования математической и информационной культуры студентов, приобретения систематизированных знаний, умений и навыков в области математической логики и теории алгоритмов, изучения ее основных методов, механизмов их развития и применения для решения научных и практических задач в области будущей профессиональной деятельности. Рассмотренные вопросы являются необходимыми элементами в современной подготовке специалистов в сфере информационных технологий.

Настоящее учебное пособие разработано для студентов, обучающихся по направлению 09.03.04 «Программная инженерия». Также будет полезно студентам направлений 09.03.02 «Информационные системы и технологии», 27.03.03 «Системный анализ и управление» и студентам других специальностей, изучающим математическую логику и теорию алгоритмов.

Методический уровень материала данного учебного пособия позволяет использовать его при самостоятельной работе студентов. Библиографический список содержит источники, к которым можно обратиться за более подробной информацией.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ахо, А. Структуры данных и алгоритмы : пер. с англ. / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М. : Вильямс, 2001. – 384 с.
2. Булос, Дж. Вычислимость и логика / Дж. Булос, Р. Джеффри. – М. : Мир, 1994. – 396 с.
3. Вайнштейн, Ю. В. Математическая логика и теория алгоритмов : метод. указания к выполнению контрольных работ / Ю. В. Вайнштейн ; Сиб. федер. ун-т, Ин-т космич. и информ. технологий. – Красноярск : ИПК СФУ, 2009. – 56 с.
4. Верещагин, Н. К. Лекции по математической логике и теории алгоритмов : в 3 ч. Ч. 2. Языки и исчисления / Н. К. Верещагин. – М. : МЦНМО, 2000. – 288 с.
5. Верещагин, Н. К. Лекции по математической логике и теории алгоритмов : в 3 ч. Ч. 1. Начало теории множеств / Н. К. Верещагин. – М. : МЦНМО, 1999. – 128 с.
6. Верещагин, Н. К. Лекции по математической логике и теории алгоритмов : в 3 ч. Ч. 3. Вычислимые функции / Н. К. Верещагин. – М. : МЦНМО, 1999. – 174 с.
7. Гаврилов, Г. П. Сборник задач по дискретной математике / Г. П. Гаврилов, А. А. Сапоженко. – М. : ФИЗМАТЛИТ, 2005. – 416 с.
8. Гринченков, Д. В. Математическая логика и теория алгоритмов для программистов / Д. В. Гринченков, С. И. Потоцкий. – М. : КноРус, 2010. – 154 с.
9. Гудман, С. Введение в разработку и анализ вычислительных алгоритмов : пер. с англ. / С. Гудман, С. Хидетниemi. – М. : Мир, 1981. – 368 с.
10. Игошин, В. И. Задачи и упражнения по математической логике и теории алгоритмов : учеб. пособие / В. И. Игошин. – М. : Академия, 2007. – 304 с.
11. Игошин, В. И. Математическая логика и теория алгоритмов : учеб. пособие / В. И. Игошин. – М. : Академия, 2008. – 447 с.
12. Клини, С. К. Математическая логика : пер. с англ. / С. К. Клини. – М. : Мир, 1973. – 480 с.
13. Колмогоров, А. Н. Теория информации и теория алгоритмов / А. Н. Колмогоров. – М. : Наука, 1987. – 303 с.

14. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М. : МЦНМО, 1999. – 960 с.
15. Кузнецов, О. П. Дискретная математика для инженера : учебник / О. П. Кузнецов. – СПб. : Лань, 2009. – 395 с.
16. Лавров, И. А. Задачи по теории множеств, математической логики и теории алгоритмов / И. А. Лавров, Л. Л. Максимова. – М. : ФИЗМАТЛИТ, 2002. – 256 с.
17. Липский, В. Комбинаторика для программистов / В. Липский. – М. : Мир, 1988. – 213 с.
18. Лихтарникова, Л. М. Математическая логика. Курс лекций. Задачник-практикум и решения / Л. М. Лихтарникова, Т. Г. Сукачева. – СПб. : Лань, 1998. – 288 с.
19. Логика. Гуманитарная энциклопедия [Электронный ресурс] В. А. Бочаров, А. С. Карпенко, С. В. Воробьева, В. С. Бернштейн // Центр гуманитарных технологий, 2002–2018 (последняя редакция: 25.08.2018). – URL: <https://gtmarket.ru/concepts/6892>.
20. Лорьер, Ж.-Л. Системы искусственного интеллекта : пер. с фр. / Ж.-Л. Лорьер. – М. : Мир, 1991. – 568 с.
21. Мендельсон, Э. Введение в математическую логику / Э. Мендельсон. – М. : Наука, 1976. – 320 с.
22. Непейвода, Н. Н. Прикладная логика : учеб. пособие / Н. Н. Непейвода. – Ижевск : Изд-во Удм. ун-та, 199. – 385 с.
23. Ноженкова, Л. Ф. Моделирование знаний : метод. указания к семинарским занятиям / Л. Ф. Ноженкова, Т. Г. Пенькова, Ю. В. Вайнштейн ; Сиб. федер. ун-т, Ин-т космич. и информ. технологий. – 2009. – 39 с.
24. Судоплатов, С. В. Математическая логика и теория алгоритмов / С. В. Судоплатов, Е. В. Овчинников. – Новосибирск : НГТУ, 2010. – 255 с.
25. Трахтенброт, Б. А. Алгоритмы и машинное решение задач / Б. А. Трахтенброт. – М. : Физматгиз, 1960. – 117 с.
26. Яблонский, С. В. Введение в дискретную математику / С. В. Яблонский. – М. : Высш. шк., 2001. – 384 с.

Учебное издание

Вайнштейн Юлия Владимировна
Пенькова Татьяна Геннадьевна
Вайнштейн Виталий Исаакович

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Учебное пособие

Редактор *А. А. Быкова*
Компьютерная верстка *О. А. Кравченко*

Подписано в печать 22.07.2019. Печать плоская. Формат 60×84/16
Бумага офсетная. Усл. печ. л. 6,9. Тираж 100 экз. Заказ № 7596

Библиотечно-издательский комплекс
Сибирского федерального университета
660041, Красноярск, пр. Свободный, 82а
Тел. (391) 206-26-16; <http://bik.sfu-kras.ru>
E-mail: publishing_house@sfu-kras.ru